

# 你好哇,程序员——漫话程序员 面试求职、升职加薪、 创业与生活

安晓辉 著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

## 内 容 简 介

《你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活》以漫画+幽默风趣的语言，形象、生动地刻画出程序员的世界。全书涵盖了程序员工作与生活、创业的方方面面，如面试、求职、升职、加薪、加班、家庭，等等。程序员的喜怒哀乐，都暗含其中，具有非常高的阅读价值。作者作为一名程序员，具有多年的从业经验，难得的是文笔出众，通俗易懂，读者读来能快速产生深深共鸣。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有，侵权必究。

## 图书在版编目（CIP）数据

你好哇，程序员：漫话程序员面试求职、升职加薪、创业与生活/安晓辉著. —北京：  
电子工业出版社，2016.4

ISBN 978-7-121-28007-8

I. ①你… II. ①安… III. ①程序设计—工程技术人员—职业选择 IV. ①C913.2

中国版本图书馆 CIP 数据核字（2015）第 321405 号

责任编辑：高洪霞

印 刷：

装 订：

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×980 1/16 印张：19.5 字数：329 千字 彩插：2

版 次：2016 年 4 月第 1 版

印 次：2016 年 4 月第 1 次印刷

定 价：49.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 [zlts@phei.com.cn](mailto:zlts@phei.com.cn)，盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线：（010）88258888。

# 为什么写

人最宝贵的就是生命，生命对于每个人来说只有一次。人的一生应该这样度过：回首往事，他不会因为虚度年华而悔恨，也不会因为碌碌无为而羞愧；临终之际，他能够说：“我的整个生命和全部精力，都献给了世界上最壮丽的事业——为解放全人类而斗争。”

我在上初中时有幸读到了《钢铁是怎样炼成的》一书，它的主角保尔·柯察金说过上面的话。在我指悬键盘准备写序言时，忽然想到了它。

从 2005 年我转行做开发至现在，转瞬之间十余年了。

十年。

作为程序员，我生命里最珍贵的十年，都献给了 Coding 。从最初的贸然进入，到后来的喜欢、热爱、坚持……一路上既有艰难困苦踟躇徘徊，也有一往无前风光霁月，有很多人很多事，经过我细小却永是流逝的生命之河，它们就像一颗颗或大或小的石子，激起了无数的浪花，漾起了经久不息的涟漪。

我曾经是一个程序员，现在仍然是，将来也许还会是。无论如何，我永远也不会忘记这天赐的身份，以及它带给我的喜怒忧思悲恐惊。

2015 年匆匆而逝，看起来与其他年份一样并无什么不同，然而回望之际，我发现，它对我来讲真是特别的一年，具有非同寻常的意义。这一年发生了很多事，无论悲喜，都给我留下了深刻的印象，并且它们纠缠交错，一刻不停地向四面八方撕扯着我，终于让我明白了一些实实在在的生活道理。

也正是 2015 年，我始终没有间断地在思考程序员、生活、理想、工作、家庭、亲人、责任、现实、担当等各种人生主题，并且终于有所收获。我不但认识到，“作为一个人，要是

不经历过人世间的悲欢离合，不跟生活打过交手仗，就不可能真正懂得人生的意义”；并且明白了，“人生的乐趣不仅在于达到某一目标的那一刻，更在于继续不断努力追求之中。在这努力追求的过程中，我们觉得生命有意义，活着有价值”。

我想通了很多事情和道理，我对工作、生活、理想、奋斗等主题有了自己的理解和看法，这些理解和看法反过来又有形或无形地指导了我的生活和工作。

因为我出了两本书（《Qt on Android 核心编程》和《Qt Quick 核心编程》），参与了技术论坛、QQ 群的一些事儿，接触了很多程序员，有刚入行的，有攒了几几年经验的，也有经验丰富学富五车的……程序员们会提出各种各样的问题，并且就这些问题在论坛、QQ 群里交流、辩论，当这些讨论涉及行业选择、技术方向、薪资、财富、目标、追求时往往没什么结果——因为这些也真真是与个体密切相关，没有一定之理。

我工作十几年，一直在中小公司摸爬滚打，经历过成长期的公司，也经历过初创型公司，做过程序员、技术专家、项目经理、部门经理、项目总监、技术总监，对各个岗位的各种事情都有所了解。我是一个爱追问爱琢磨又自以为是的家伙，自以为想明白了很多道理，而这些道理也许是别人——比如在门外观望或新入行的兄弟姐妹们，比如干了几多年仍然不断面临各式各样迷惘困惑的老手，比如像我一样自以为想明白了一些人和事需要别人加以印证的伙计所需要的……所以，我开始写作“漫谈程序员”系列文章，并在博客上保持一周一篇的节奏持续更新。

没有想到的是，这些文章居然颇受关注，不但在 CSDN 网站点击量很大，还被大大小小的各类技术相关的站点转载，这是我始料不及的，内心有一丝高兴也有一丝惶恐。高兴的是“有这么多人的想法与我相似，那我领悟的这些鸡零狗碎看来真可以帮到一些人”，这正是我的初衷；惶恐的是，如果我错悟了某些事，通过文章传达了出去误导了别人，可就罪莫大焉了。

我就在这种高兴与惶恐中前行，一直走到了现在。

涓滴之水终可磨损大石，不是由于它力量强大，而是由于昼夜不舍的滴坠。我相信我的坚持一定是有价值的，起码我在做我想做的事情，我不会感到无聊。如果正好我做的事能对别人有所助益，帮助别人发现自己、找到他们想做的事并投入进去，那我不仅不会无聊，还会感到欣慰，因为我这有限的一份热一份光，毕竟有了自我之外的一点价值。



## 致 谢

感谢本书的策划及文字编辑高洪霞老师，她用一罐子热情洋溢的肯定打消了我出这样一本书的各式各样的担忧，萦绕我心头的那些“不务正业”、“哗众取众”的声音在高老师持续的鼓励下终于慢慢飘散。同时高老师对我的信任和支持也让我能够有充足的时间在相对自然的状态下慢慢思考、写作，这也使得本书中的大部分文章在最大程度上保留了源自程序员生活的原汁原味。

感谢出版社以及本书的各位编辑，没有你们的辛苦付出和鼎力帮助，本书绝难顺利付梓出版。

特别感谢本书的插画作者“不知霜舞哀伤”，一面之缘也无，她居然能信任我，答应为本书配图，让我温暖不已。幸何如之！

感谢我的老婆赵建媛女士，她不但在生活上给予我支持，还是我忠实的读者，她不断的催促我在博客上更新，她的期待、鼓励、建议，是我一路坚持的强大动力，没有她的理解和宽容，我的一切都是梦幻泡影。

感谢我的女儿安悦瑄小朋友，我在《Qt on Android 核心编程》中提到她时，她四岁半多，在《Qt Quick 核心编程》中提到她时，她五岁了，现在她过了五岁半，一阵风似地往六岁跑去了，她天真的期待和向往让我觉得自己在做的事情有了别样的意义。

特别要感谢的是在 CSDN 上追看我“漫谈程序员”系列博文的朋友们，我从博文后面那些热情、同感的回复以及三不五时收到的私信中汲取了满满的能量，被需要、被认同的感觉是我能够坚持更新的最大动力，没有你们，就没有我的“漫谈程序员”专栏，也没有本书，你们是最可爱的人！

感谢已故的著名作家王小波，他既是文坛外高手，也是程序员中的奇葩。“你好哇，程序员”这个名字受到他写给李银河女士的情书的启发，特此感谢。

感谢我昔日的 BOSS，他们给我机会让我成为程序员，曾经一路走过八年时光，甘苦毕生不忘。他们是谢胜落、韩军。

谈笑有程序，往来无白丁，感谢曾经经过和正在经过我程序员之路的同事、朋友，我们在一起的日子，若隐若现在扯淡的文字里。

感谢 CSDN，感谢 CSDN 博客频道的编辑们，正是有了这个平台和你们辛勤的劳作，我的一切才有可能。

最后，也最应该感谢的，是肯花费时间和精力阅读本书的你，没有你，我将寸步难行。

这本书适合谁

这不是一本严肃的、正儿八经的谈论程序员的书，它只是我这么一个老程序员在工作、生活中撷取的几朵小浪花，谈不上多么深刻，也谈不上多么有趣，但，它是真实的。所以，如果你：

- 是一个程序员
- 想成为一个程序员
- 想了解程序员的工作与生活
- 对程序猿这种特殊物种感到好奇

都可以从本书中找到你想要的东西。

如果你只是想吐槽程序员，也可以从本书中找到各式各样的靶子。

# 目录

群星闪耀的黄金时代 /	1
3D 打印能打印出“程序猿”吗 /	4
程序员到底是什么角色 /	11
那些你珍视的，却被别人视为垃圾的 /	15
程序员零门槛吗 /	20
漫谈选人与培训 /	23
自我发现，找到适合自己的职位 /	26
亲爱的“程序猿”们怎么找工作 /	29
找工作的辟邪剑谱 /	34
薪资，你是我不能言说的伤 /	43
程序员该不该考虑初创公司 /	55
任性，春节前辞职 /	63
让程序员跳槽的非钱原因 /	67
无 Bug 不生活 /	73
谁是为加班而生的 /	84
别说“我已经很努力了” /	89
程序员的生活就这样吗 /	95
软件开发的十八般乐趣 /	106
你的幸运女神呢 /	114
那些害死程序员的细节 /	120

千奇百怪的程序员 /	129
咦，你也在混日子啊 /	137
让程序员无语的那些事儿 /	143
多任务的神话 /	149
乐观的程序员 /	154
怎么告别“混日子” /	160
一个老程序员的 2014 /	171
坦承无知有多难 /	177
就这几点，提高工作效率 /	182
程序员的第二生活支点 /	189
程序员的年龄怎么着了 /	195
不要让自己成为问题 /	204
怎样才能多读一些书 /	211
伤心小箭，你中了几枝 /	217
什么仇什么怨 /	223
看看你离优秀有多远 /	239
问题去哪儿啦 /	246
怎样成为技术达人 /	253
一张图道尽程序员的出路 /	262
技术人员为什么要写博客 /	269
程序员和女朋友（男版） /	272
程序员和女朋友（女版） /	281
生个孩子吧 /	288
推荐给程序员的书 /	300

## 群星闪耀的黄金时代

豪杰超级解霸是 1998 年上大学时的最爱，当时学校还没有互联网，东洋和海外的诱惑很大，一群光棍的看碟需求很强烈，超级解霸啊，视频播放江湖的绝对霸主，它的作者，就是大名鼎鼎的梁肇新。一个程序员搞定一款风靡江湖一时无俩的软件，还由此创办了北京世纪豪杰计算机技术有限公司，不能不说这是传奇。当然，还有更传奇的程序员，咱们先看看他们的辉煌战绩，YY 一把黄金时代。

说到黄金时代，我想先说说《黄金时代》这本小说，它是已故的著名小说家王小波的大作。王小波的夫人是李银河，一对牛人！

王小波这个人，不但是一个优秀的小说家、杂文家，特立独行的自由骑士，还是一个非常牛气的程序员！

当时还是 286 的年代，为了方便自己处理文字，他开发了中文编辑器和输入法，想想看，一个人为了自己的需求在缺乏可参考样本的情况下开发了这么复杂又深具开创性的软件，这是多么不可思议！有多少人找他搞计算机搞软件赚钱，而当时这个也真是赚钱，可是他拒绝了，还是决定写小说！

说到 DOS 时代的中文处理，还有几个牛气冲天的程序员，比如写出 CC-DOS 的严援朝，开发 2.13a 中文系统的吴晓军，希望汉字系统 UC DOS 的开发者鲍岳桥和简晶，五笔字型的王永民，这些人就像天上的星星亮晶晶，照亮了一个独行侠的童话时代。



开发 WPS 的求伯君、雷军，大家可能更熟悉，而且雷军现在更是无人不知、风头超劲。“小米”虽然没有步枪，可也不是一般的强，2014 年曝光率奇高，几乎造就了十万个冷笑话的大蓝翔就是再喊“挖掘技术哪家强”也难望其项背。

Windows 的出现终结了 DOS 时代这些独行骑士般的程序员创建的辉煌，可是还有新的传奇，比如文章开始提到的梁肇新和他的豪杰超级解霸，以拓荒者的姿态一统视频播放江湖。宽带慢慢开始普及时，大名鼎鼎的自由软件网际快车（FlashGet）以及它的作者侯延堂，给程序员展示了共享软件之路可以如何辉煌，它在全球有超过 1 亿的用户，66 种语言版本，是当时装机必备！

我非常崇拜这些程序员中的程序员，对他们的景仰之情难以言表，想到他们，已经灰暗了几十年的小眼睛也会一闪一闪地放点光出来。

这都是不朽的传奇啊！

还有一位最富传奇色彩的程序员：王江民。

王江民 38 岁才接触计算机，43 岁时写出杀毒软件 KV！请别告诉我程序员只能干到 35 岁啦……

KV 是畅销千万的系列软件，笑傲江湖多少年！

KV50，KV100，KV200，KV300，KV300+，KV3000，KV2004，KV2005，KV2007……

KV2007 是在人民大会堂发布的！

时间到了 1995 年，微软推出温酒屋（Windows 95），中国通用软件产业格局由此改写，DOS 平台的中国软件骄傲 WPS、UCDOS、CCED 转瞬成灰。写出中文之星的廖恒毅进了微软，求伯君和英雄少年雷军率领的 WPS 也被 Office 打成了低首神龙……

可是王江民崛起了，逆流而上成功破局。

王江民的传奇，还在于他开启了一个程序员也可以发达的时代，给千千万万程序员树立了一面梦想的旗帜。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

“大丈夫当如是也”，这是刘邦见始皇帝之风光后说的话，这也是千千万万苦逼程序员仰视王江民时心中所想。

2010年4月4日，王江民病逝……一代程序员的背影也渐渐地看不见了，程序世界的个人英雄主义时代当然也早杳如黄鹤。

可是，程序员的平凡之路还在继续。无论这个时代多么依赖和提倡集体协作，但个性的张扬永远不会泯灭，永远让人激动不已，因为它代表着个人存在的价值和意义。假如有时你陷入了“不知何事萦怀抱，醒也无聊，醉也无聊，梦也何曾到谢桥”的低徊之中，就让我们仰望星空，怀想黄金时代里驰骋江湖的英雄吧，那些传奇，会让我们心潮激荡，温暖由此而生，力量在一圈圈涟漪中集聚，在这程序猿们悲鸣的黑铁时代里，纵然中关村不再相信传奇，纵然个人英雄主义已成过去，可独特的你，依旧可以在“我是希瑞，赐予我力量吧”的叫声中爆发，走出属于你个人的独特历程。

## 3D 打印能打印出“程序猿”吗

### 3D 打印的定义

关于 3D 打印，下面是来自百度百科的解释：

“3D 打印，即快速成型技术的一种，它是一种以数字模型文件为基础，运用粉末状金属或塑料等可黏合材料，通过逐层打印的方式来构造物体的技术。”

“3D 打印通常是采用数字技术材料打印机来实现的。常在模具制造、工业设计等领域被用于制造模型，后逐渐用于一些产品的直接制造，已经有使用这种技术打印而成的零部件。该技术在珠宝、鞋类、工业设计、建筑、工程和施工（AEC）、汽车，航空航天、牙科和医疗产业、教育、地理信息系统、土木工程、枪支以及其他领域都有所应用。”

### 3D 打印的传奇

我们先来看看 3D 打印闪光的历史吧，特摘取几个激动人心的事件。

### 3D 打印汽车——斯特拉迪

2014 年 10 月 10 日，3D 打印汽车出现了！

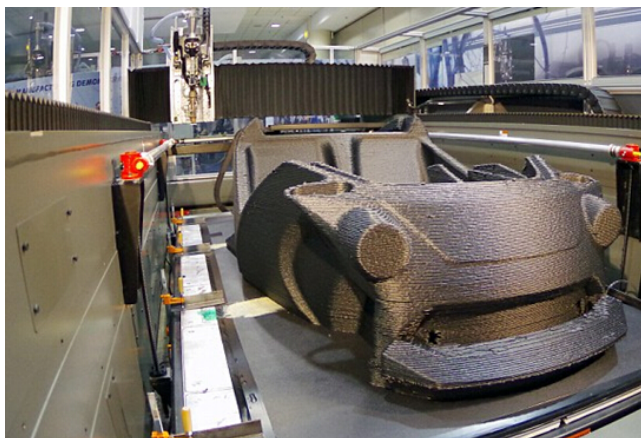
新闻标题为“纽约街头惊现世界首款 3D 打印汽车 仅 44 小时完成制作”，链接为 [http://www.chinadaily.com.cn/hqgj/jryw/2014-10-10/content\\_12505608.html](http://www.chinadaily.com.cn/hqgj/jryw/2014-10-10/content_12505608.html)。

这是全球首款 3D 打印汽车，“斯特拉迪”，由“本地汽车”公司打造，整辆汽车的零件成本约为 3500 美元，制作周期为 44 个小时，并且最高时速可以达到 80km/h，最不可思议的是，这还是一辆电动车，充电一次 3.5 小时，可以行使大约 100 公里。

看一些图片吧：



🗣️ 你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活



3D 打印汽车



3D 打印汽车上路了

## 3D 打印房屋

我女儿非常爱看我和媳妇推荐给她的《鼹鼠的故事》，其中有一集是小鼹鼠和它的小伙伴们所生活的森林被强拆了，摩天大楼平地起，那些房子，都是从别处运来的，像搭积木一样搭起来，一会儿工夫几十层的高楼就像糖葫芦一样插满地面。卫生间也是如此这般……当

时我就想，房子这么盖，真牛！

后来我看到了一则题为“3D 打印狂人马义和”的新闻：

“2014 年 3 月 29 日，上海盈创装饰设计工程有限公司在同济大学逸夫楼举办建筑 3D 打印发布会上宣称，公司在苏州厂区内有一台高 6.6 米、宽 10 米、长 150 米的 3D 打印机，可以在 24 小时内打印出 10 栋 200 平方米的水泥平房，这些可以供人居住的房子正在上海张江高新青浦园区内的一片空地上展示。

“消息甫出，国内 3D 打印界一下子沸腾了。此前，发达国家也曾推出过 3D 打印房，但材料要么是塑料，要么是水泥，建造速度不如盈创那么快，也从未见过盈创宣称的那么大的 3D 打印机。

“上海盈创装饰设计工程有限公司董事长马义和自豪地向《新民周刊》表示：专注生产绿色环保建筑材料 12 年的盈创公司可以连续 3D 打印整栋房屋，无论是低层还是高层；而且 3D 打印的“油墨”是建筑垃圾的再利用，包括工业垃圾、尾矿等，既降低了建筑成本，又减少了对环境的破坏。”

新闻链接在这里：<http://www.xinminweekly.com.cn/news/content/3568>。

这真让人激动啊……然而谁能告诉我新闻是真是假？

## 3D 打印的历史发展

以下内容摘自百度百科：

- 1986 年，Charles Hull 开发了第一台商业 3D 印刷机。
- 1993 年，麻省理工学院获 3D 印刷技术专利。
- 1995 年，美国 Zcorp 公司从麻省理工学院获得唯一授权并开始开发 3D 打印机。
- 2005 年，市场上首个高清晰彩色 3D 打印机 Spectrum Z510 由 Zcorp 公司研制成功。
- 2010 年 11 月，世界上第一辆由 3D 打印机打印而成的汽车 Urbee 问世。

☞ 你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

- 2011 年 6 月 6 日，发布了全球第一款 3D 打印的比基尼。
- 2011 年 7 月，英国研究人员开发出世界上第一台 3D 巧克力打印机。
- 2011 年 8 月，南安普敦大学的工程师们开发出世界上第一架 3D 打印的飞机。
- 2012 年 11 月，苏格兰科学家利用人体细胞首次用 3D 打印机打印出人造肝脏组织。
- 2013 年 10 月，全球首次成功拍卖一款名为“ONO 之神”的 3D 打印艺术品。
- 2013 年 11 月，美国得克萨斯州奥斯汀的 3D 打印公司“固体概念” (SolidConcepts) 设计制造出 3D 打印金属手枪。

那么问题来啦

3D 打印机能不能打印出“程序猿”？其实有的“程序猿”很关心另外一个问题：3D 打印机能否打印出女朋友？抱歉，这个问题不在讨论之列。为了论证 3D 打印机到底能不能打印出“程序猿”来，我们得先看看“程序猿”到底是何方妖怪。

我将从“程序猿”的定义、外观及社会形象、物质组成三个方面来说明“程序猿”到底是什么。

## “程序猿”的专业定义

还是看看百度百科对“程序猿”的精确解释：

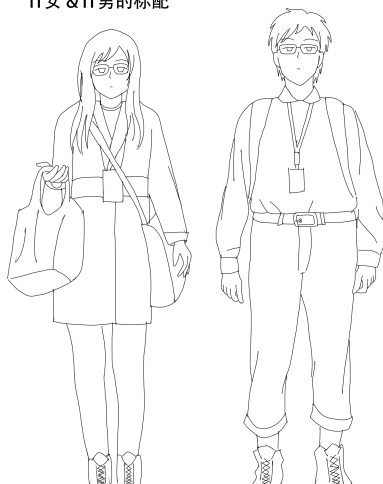
程序猿是一种近几十年来出现的新物种，是信息革命的产物，在行为和物种归类上也可称为码字猴。程序猿是人类在科技研究上的一种新兴进化，拥有无与伦比的耐力、超越时代的智商、横穿社会的苦逼相和低于人类平均寿命的显著特点。

程序猿是一种非常特殊的、可以从事程序开发、维护的动物。一般分为程序设计猿和程序编码猿，但两者的界限并不非常清楚，都可以进行开发、维护工作。特别是在中国，最重要的一点，两者都是一种非常悲剧的存在。

3D 打印能打印出“程序猿”吗.🐼

## “程序猿”的外在形象及社会形象

IT女 & IT男的标配

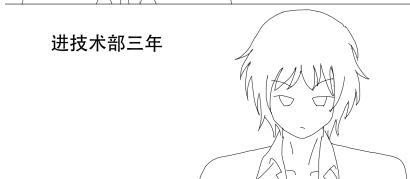


“程序猿”的进化史，敢问路在何方，敢问今夕何夕：

刚进技术部



进技术部三年



进技术部五年



图 3

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## “程序猿”是何种有机体

通俗地讲，“程序猿”，实际上是程序员们的自嘲。

构成人体最基本的元素有几十种，最多的是水，此外还有的蛋白质等。

换一种感性的说法。

肉体+灵魂+思想，组成了人。

虽然肉体+灵魂+思想可以组成人，但是肉体+灵魂+思想并不是就会等于人，肉体+灵魂+思想也可以等于其他生命。

人的肉体是由各种物质组成的，可以再造。但是灵魂和思想是基于肉体和社会认知活动发展出来的，很难再造。如果你相信孟婆汤和奈何桥，那么一定要记得，过奈何桥时不要喝那碗孟婆汤，以便你投胎转世时可以拥有前世的灵魂和记忆，这样就可以像愚公移山那样，子子孙孙世代代来成就伟业。

作为一个“程序猿”，编程技巧、设计思想、项目管理、产品思维等，都是后天发展出来的。你看逍遥子这种牛人，能够传功给虚竹，可是其潇洒出尘风流倜傥之卓越丰姿，还是无法传递给虚竹这个形容粗陋的小和尚的，这些东西是天生身体条件加上后天发展培养而来的，没有办法一股脑地转移给另一个人……

可是这些和 3D 打印机有关系吗？

## 3D 打印与生命体的绯闻

### 1. 较早的打印人体器官的新闻

看这里，来自网易的题为“美国科学家成功 3D 打印人体器官”的新闻：

“英国《每日邮报》报道，美国维克森林大学的科学家成功利用 3D 打印技术打印人体细胞的生物可降解支架，而后利用支架培育耳朵和鼻子等仿真度极高的面部器官。利用 3D 打印技术培育用于移植的肺部以及其他器官是科学家的终极目标。科学家表示他们在实现这一

目标道路上正不断取得进步，有望在 20 年内让患者移植 3D 打印器官的梦想成为现实。”

看来，人造器官终究有一天会实现啊。

欲查看原文的，参考链接在这里：<http://discovery.163.com/13/0619/11/91NQEAMS000125LI.html>。

## 2. 美国医生用 3D 打印心脏救活 2 周大先心病婴儿

网易 2014 年 10 月 13 日的新闻“美国医生用 3D 打印心脏救活 2 周大先心病婴儿”。摘要如下：

“据外媒报道，利用 3D 打印技术可以改变人们生活，之前就有许多关于 3D 打印义肢帮助患者恢复正常生活的报道。纽约长老会医院的埃米尔·巴查博士（Dr.Emile Bacha）医生就讲述了他最近使用 3D 打印的心脏救活一名 2 周大婴儿的故事。”

原文地址：<http://news.163.com/14/1013/09/A8E7DORR00014JB6.html>。

既然人造器官是可能的，核心部件心脏都打印出来了，那看来 3D 打印出一个大活人也是有可能的啦！真叫人难以置信，我竟然得出了这么个结论！

好在，还有下文。

这里让我们谈谈肉体之外关乎灵魂的事儿。抱歉，我虽然是理工男，可是这些东西也是科学，是可以归纳证明的。

## “程序猿”不可复制的独特性

每个人都有独特的味道和灵魂，这是人之所以区别于他人的关键。而“程序猿”是人，所以必然有自己的不可复制性。3D 打印能够根据模板公式打印出具体物件，可是“程序猿”是有灵魂的，灵魂不可复制，所以 3D 打印只能打印出“程序猿”的肉身，打印不出一个落地就能编程就能设计复杂软件系统的怪咖！

那么，如果 3D 打印出了很多貌似“程序猿”的行尸走肉，跑到公司来抢“程序猿”的饭碗，怎么办？其实，这些貌似“程序猿”的假“程序猿”，终有一天会显出原形，也会消失的。

## 程序员到底是什么角色

先讲个生活小插曲儿……

### 瓦工

房客打电话给我，说卫生间漏水，把楼下的客厅都洇湿了，老先生找上门来了。我心想这还了得，赶紧处理吧。我找了人拆掉了浴缸，砸掉了地板，买了东方雨虹的防水涂料，自己做了防水，闭水试验，最后一步就是贴瓷砖了。贴砖是个技术活儿，很专业。我是程序员，虽然可以勉强刷个防水，可贴砖实在干不了。在 58 同城找了一通电话，因为卫生间面积太小，又是改造，所以有的瓦工不愿意干，愿意干的瓦工所开的价钱我又不能接受，十多个电话过去后才终于找到一个师傅，愿意以 400 块钱把这活儿拾掇了——这还是因为那两天下雨，没别的活儿可干。于是我冒雨买了水泥、沙子、瓷砖，陪着杨师傅把地砖和几十公分高度的一圈墙砖给贴了。

杨师傅是这么裁砖的：拿玻璃刀划拉一下，在划痕下方垫一个铁钉，双手按住瓷砖两边，猛劲向下一按，啪一声脆响，砖就分成了两半，断口齐整如镜。我看着好轻松啊。只有需要斜切，或者裁掉几公分时才会用到电动的切割工具。一旦用电动的切割工具裁砖，灰尘就会更多，眼睛也会睁不开，同时噪声也很大，让人难以忍受。此时我只好躲到门外去。

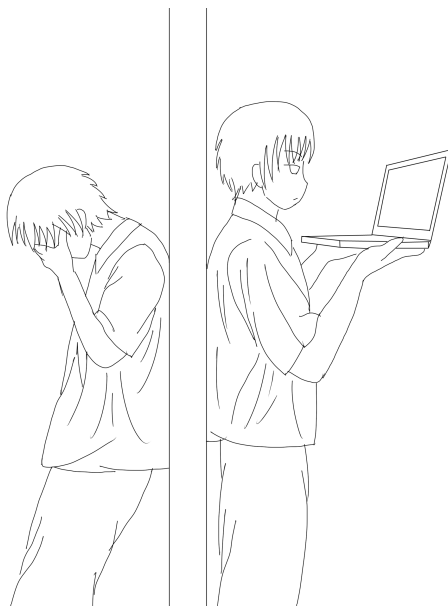
找平和做斜坡是技术活，不仅需要用水平尺测量，也需要设计。如果一不小心地漏处成不了最低点，那洗澡水就无处可去了。

干活的间隙我们抽烟聊天，杨师傅说平时接活的机会很多，我这一单是他在 58 同城接的第一单，平常都是熟人推荐，基本上没有闲着的时候，平均一天 500 块。我一想这是纯收入，比很多程序员高多了，甩他们几条街啊。

我之所以有时间忙活卫生间改造的事情，是因为我刚好失业，不用上班不用打卡。而上班的日子，一点儿事儿都要跟领导请假，打卡晚 5 分钟都算迟到，既要扣全勤还要罚款。我这程序员当得不自由啊！而瓦工则没有这方面的限制。杨师傅约好八点到，八点半了还没见

人影，我也不能少给人家 100 块钱。其实程序员也需要自由，公司真不用限制那么死，只要做好派活儿、验收两件事就可以。可实际上很多人陷入了项目经理的微管理中。

## 程序员



有一阵子网上疯传“小明和小强的故事”，矫情的白领们纷纷感叹自己还不如外出务工的农民兄弟。很多程序员也在恶搞自己的身份，PS 建材市场附近的务工者图片，把牌子上的字样改成“C++”、“Java”、“.NET”等。

可实际上，有几个程序员愿意转行去做瓦工、水电工、木工呢？就是车工、钳工、氩弧焊恐怕也不愿意吧？

钱不是唯一的差异，在钱相差不大的时候，社会身份、工作环境是更有份量的因素，甚至它们也可以让你接受相当程度的薪水差异。我跟着杨师傅贴了一天的砖，我只能说，这活儿不是人人都能干的，一蹲几个小时，我这胳膊腿早麻了；那么大的粉尘，我这鼻子、喉咙、



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

肺早完了。这是个手艺活儿，却也是体力活儿，辛苦活儿。

瓦工、木工、水电工，都是匠人。程序员说白了也是匠人，写程序也是个手艺活儿。

早在我上高中的时候，我的父亲就劝诫我将来学一门技术。在他的观念里，无论世界怎么变幻，技术都是立身之本，一技傍身，到哪里都不愁吃喝。等我考大学的时候，他不能理解信息工程这种专业也是教技术的，等我干了程序员他也不能理解编程是项技术活儿。可现在他理解了，认同这是一门手艺。

我的老板跟我说，说到底软件开发是手艺活儿，到什么时候软件都不可能自动生产出来，软件写得好坏全靠个人手艺。没错，我认为这是正理儿。这也是我对程序员这个社会分工的看法，乌泱乌泱的程序员，和瓦工、木工、教师、电工、保洁工等没有太大的差别，只是分工不同而已。每一行都有自己的欢乐忧愁，而隔行如隔山，我们能看到的，仅仅是数字化的报酬，我们程序员矫情的感叹，往往也只是一闪念，谁也不会当真，当真就输了。

## 别人怎么看程序员

首先我们得搞明白自己怎么看自己，如果你觉得自己干的活儿没有意义甚或卑贱，那别人再怎么给你高帽子你都高兴不起来。最重要的是自己对自己工作的看法，不要去管别人说三道四。大风吹倒梧桐树，自有别人论短长，怎么论是别人的事。

确认了这一点，我们就可以看看程序员在别人眼中的样子了。

有时别人认为程序员是修电脑的——我曾经给很多人修过电脑，这是真的。

有时别人以为程序员是坐办公室的，工作很轻松，不需要日晒雨淋。我的那些务农的乡亲们就是这么看我的。

有时别人以为程序员是白领，薪水高。可白领到底是个什么定义呢？

有时别人认为程序员就是喝咖啡产代码的怪咖，不通人情世故。我要说，兄台，我们有时也抽烟，更多的时候拎着豆浆、包子、煎饼果子、地沟油炸的油条匆匆来去，咖啡其实算

是奢侈品。

有时别人认为我们就是码农。

有时别人以为我们就是专职给用户添堵找罪受的。

有时别人以为我们是一群神秘的骇客，噼里啪啦一顿乱敲就侵入了银行系统，可以偷偷给自己的户头转上一笔钱。或者我们可以左右双色球的开奖结果，帮助他们中一等奖，而且是倍投。

有时我们根本不存在，存在的只是软件。

……

其实这都无所谓。

我们给进城务工的农民兄弟起了个名字叫作“农民工”，你可以在拥挤的人潮中迅速辨认出农民工。没错，他们有他们的特殊印记。实际上，程序员也有自己独特的身份标记。我学开车那会儿，和一哥们儿聊天，他上来就问我你是不是搞软件的。我很诧异啊，他怎么知道？其实你也可以在大街上看出来哪个人是当兵的，哪个人是老师……只要你注意观察，每一个行当都会给人留下特殊的习惯、气质印象，我们从事的工作在无形中改变着我们，随着时间的流逝，我们最终会因为工作而成为一个特色鲜明的群体。

古代有一种墨刑，又叫黔面。施行的方法是在人的脸上或身体的其他部位刺字，然后涂上墨或别的颜料，使所刺的字成为永久性的记号。这种刑罚不仅给人造成肉体上的痛苦，同时使人蒙受巨大的精神羞辱。如果你看过水浒传，一定会对林冲、武松等印象深刻，他们脸上都有字。甚至连名传千古的狄青，脸上也有字。墨刑给人制作了一张永久性的名片，是贴标签的极端体现。而贴标签这种事情，我们太擅长了，它简单好用，效果显著，所以常常被拿来标注别人，我们也常常被别人这么标注，比如说这个人，是好人那个人是坏人。

好吧，最好的办法是让一切都随风，心随意动。

## 那些你珍视的，却被别人视为垃圾的

每个人都是独特的、与众不同的，每个人又都是社会化的，生活在关系之中。在我们自我完善的途中，会有很多事我们视若珍宝而别人目为垃圾，这是个体与群体、个体与他人的差异造成的。然而我们要成长、成熟，就会不断经历这样的过程。

其实这个题目，来自于韩寒策划的“一个”（APP）某一期的问题。

所以，一个对纯文学有追求的赤子可能会耻笑我居然会看“一个”这种没营养的APP，其实，这正是我的题目所言。

### 闲X

记得我小时候特别喜欢看书，我在“怎样才能多读一些书”一节中也有提及，那时我最爱看的是武侠小说，从小学四年级看到高三。

我高中时所在的班级，大部分同学是通过奥林匹克数学竞赛保送进去的，还有一部分是因为中考成绩优秀进去的，总之兄弟姐妹们要么是偏科的怪咖，要么是学霸，个顶个的聪明。

聪明人很容易被学习之外的事情吸引，比如武侠、言情小说那时就比较受欢迎，还有那时流行的电脑游戏，比如红警，也很受欢迎。

我们的数学老师特别负责任，现在提到了，一定要表示感谢，如果没有负责的老师，不知有多少人会被挤下独木桥去。但当时，因为很多同学都觉得自个儿聪明，老师讲的内容又常常比较基础，自认为都会了，加之武侠世界真真精彩，情节跌宕起伏不说，主角还总奇遇连连美女常伴，实在是遏制不住快快看完的念头，于是就偷偷把书藏在桌子下面或者课本下面看，以为老师不知道。其实呢，你自己往讲台上站一下试试就知道了，老师不可能看不见你的小动作。作为老师，其实他的选择题就两个选项：收拾你还是不收拾你。而我们的班主

那些你珍视的，却被别人视为垃圾的🐼

任，通常会选择“收拾”这个选项。于是，我们就只好把书上交，检讨，然后赔钱给校门口的租书店。

作为一个程序员，模棱两可的说法是应当被摒弃的，我们奉行的是直话直说、简单生活、简单爱、简单的快乐御宅的自在。所以，我要说的是，这些事其实没有对错，有的只是白天不懂夜的黑。

我想说的，当然不仅仅是看别人眼中的闲书，而是“兴趣”。所以，游戏、德州扑克、画画、养鸟、种花，等等，都具有相同的性质。

## 梦想

又来谈梦想了。

梦想还是要有的，万一实现了呢。多好的一句话啊。可有些人见不得别人梦想成真。

那些不愿努力安于平凡的人会千方百计打击你的梦想，因为梦想对他们来说太残酷了，梦想会刺痛他们，让他们觉得不舒服，他们要让你和他们一样，因此会想尽办法告诉你，你的梦想都是空谈、是垃圾，要多愚蠢有多愚蠢，直到你放弃了，他们会说：看，我说了吧，早该……

平庸的人希望你和他们一样平庸，有时仅仅因为你优秀就会成为众矢之的，所谓木秀于林风必摧之，大抵是此道理。

但是，我们不可以放弃！

来个小段子，看看程序员的魔力北极光：

“‘程序猿’问‘程序媛’：‘为什么要离开我，我做得还不够好吗？’‘媛’说：‘别傻了，我们根本就是两个世界里的人，就像在 JS 里永远都无法调用 Java 类一样，我们之间也是不可能的。’‘猿’沉默了很久，转身离开了。一个月之后，他在开源社区公布了 DWR（Direct Web Remoting）的完整代码。”

## ☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

好的东西往往是难以描述的，你的梦想之所以不被支持，是因为那些不支持你的人，已经失去了做梦的能力，难以辨识好东西，他们和你不在一个频道上，无法理解你变不可能为可能的“神力”。

如果没有梦想照进现实，现实该是多么平凡、琐碎、缺乏诗意啊。人总得有一件合理的事去追求去做，才会有归属感，才能感到存在的价值，才会有幸福的生活。

我们要关心自己的感觉，不要活在别人的看法里。所以，请走自己的路，让别人说去吧。

### 自己

你很重视自己，认为自己才华独具，即便眼前还不能展翅高飞，你也相信不久会一飞冲天，即便不能一飞冲天，也可能收获不一样的人生。可是有人见不得你好，怕你好，会把你说得一文不值，直接的会说——“就你？还想……”，间接的会说——“人要脚踏实地，知道自己能干嘛！”

但是我们一定要明白，人生最本质的财富，是你自己，你自己就是一座巨大的矿藏，只要开发，就能有无穷的潜力。也只有开发，你的一切才能显现出来，才能熊熊燃烧，才能闪出光彩。

面对别人的怀疑、否定、讥讽，我们只需要坚定地前行，做给他看。行动是最好的证明，多坚持一秒，就会不同。

其他任何人都可以放弃我们，因为他们不必对我们负任何责任；唯有我们自己，决不可以放弃自己，因为只有我们自己才能对自己负责任。

### 女朋友

有时别人很难理解你为什么会找那么一个女朋友，可是，你爱她啊。

那些你珍视的，却被别人视为垃圾的🐼

因为“我爱你，不光因为你的样子，还因为，和你在一起时，我的样子”。

这是别人所不能理解的，也没办法理解的。所以，他们只能从外在的、可见的、能看懂的、可以评论的东西来评判这个女朋友是否合适。

可是，鞋子合不合脚，穿上去舒不舒服，是只有自己最清楚的。

所以，如果你爱了，那就爱了，不懂你的人才会嫌弃你的女朋友，子非鱼，安知鱼之乐乎！

## 代码

终于说到程序员的宝贝了！

大多数程序员都珍视自己的代码，无论它稚嫩、成熟，无论它简洁、复杂，我们都很爱它。可是别人就不一定这么看了。

假如你的老代码被别人维护，很可能就会被当成垃圾，有的人甚至会像掸去桌面的灰尘一样摒弃你的代码，用他们的代码来重构整个系统，抹去你曾来过这个世界的痕迹，然后还要说，“什么烂代码，简直没法看”。

## 老程序员的经验

都说程序员吃的是青春饭，所以，当你人老珠黄美丽不再，老板就开始嫌弃你了。一大波年轻力壮能加班能熬夜的小鲜肉正在赶来，谁还在乎你呢！



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 准程序员的年轻

如果你从别的行业来，想做一个程序员，你就会面临这样的窘境：我们要求工作三年以上，有相关项目经验，精通 Java、Struts、Spring、Groovy、MySQL、JavaScript、PHP、Perl……

你说你年轻，可以学习，可是人家就想开箱即用。

你说你年轻，能加班能吃苦，可是人家就想立马解决问题。

## 程序员零门槛吗

6 个小故事，一一来看。

### 技术支持转开发

2005 年 3 月我决定辞去技术支持工作，转行做软件开发。

大学时我曾经学习过 PASCAL 这种编程语言，但和程控交换机打了几年交道之后，PASCAL 早已灰飞烟灭，渣都找不到了，而且我孤陋寡闻，也没听说哪个软件公司用 PASCAL 做开发。于是呢，我决定学习 C 语言，花了一个星期，走马观花式学习了由 Brian W. Kernighan 和 Dennis M. Ritchie 合著的《The C Programming Language》(译作《C 程序设计语言》)，然后就开始了我长达一个多月的求职之旅。

现在我对那段经历还记忆犹新，主要是被鄙视的次数较多。一个星期，仅仅能够了解到基本语法吧，而且凭着五分熟都不到的一门语言就贸然踏上求职之路，不是我太自信，而是太天真。经历了两个多星期不断的笔试，我已经能够通过很多公司的 C 语言笔试题目了，真让人高兴。可是到面试时，经常会被问“你做过什么项目”、“写过多少行代码之类”的问题，面试官探究的目光和掩藏的失望与不屑，让我既无奈又伤心……不到 3 个星期，起码投了几十家公司，笔试了七八家，面试了三四家，都没有通过。

当时也曾去华为面试过，到三面的时候让我去慧通，我拒绝了。那是走得最远的一次。

后来我在软件园找到一家小公司，过了笔试关，过了“一面”，到研发副总面前面试，没怎么聊编程的细节，畅谈了我过去的工作和各种爱好。然后他给我一个星期，让我用 C 语言解决马踏棋盘问题，后来我解决了，就进了这家公司，此时已经 5 月份了。

### IBM 的专家，原来是卡车司机

我爱人曾经在 SPSS 工作，SPSS 被 IBM 收购后，她就成了 IBM 的员工。她和她的同



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

事经常与美国那边的同事沟通，有一次一个专家到西安来，聊天时了解到，他以前是开卡车的，后来转做开发，现在是该领域的资深专家。具体细节我就不太清楚了，这里也无法细述。

## 我的女同事从测试转岗开发

我有个女同事，一直做测试，2013 年的时候忽然干起开发来，我吃了一惊。因为我这人不大会八卦，消息闭塞，不知道个中缘由。

在国内，测试不受重视，工资和开发人员相比差一大截。这也是有些测试工程师想转做开发的原因吧。我那同事就是这样，相信私底下一定花费了不少精力来学习 Python 和 Java。总之在我的印象里，她比从前瘦了不少。

## 腾讯有个从保安岗位转来的开发

“腾讯保安华丽转身 IT 精英 马化腾称其故事励志”，这是来自腾讯的 2012 年的新闻。摘要如下：

“从保安到 IT 精英，中国也有林书豪一样的励志故事。昨日微博上有消息称，腾讯北京分公司 20 楼前台一名保安经过层层面试被腾讯研究院录取，腾讯公司方面证实了该消息，公司董事会主席兼首席执行官马化腾也通过微博称其故事很励志。”

当然这名保安在被曝新闻时还不是程序员，但这无关要旨。

## 张慧华的故事

张慧华，浙江金华人，1979 年出生。主要从事 HTML5 手机 App 开发、Java 开发、前端开发和 PHP 开发，DWZ 富客户端（JUI）创始人之一。从事的互联网开发工作包括电子商务企业建站平台、房地产信息平台、绿色建筑能效评估平台、在线阅读器、DWZ 富客户端框架（j-UI）和 HTML5 手机 APP。

1996 年，17 岁的张慧华从厨师班毕业，做了三年厨师，1999 年开了自己的饭店，接触

到计算机，一见钟情，发现自己对此兴趣浓厚，1999 年底转让了自己的饭店，来到北京中关村做电脑 DIY 工作。后来上了北理工的夜大，参加了几个培训，转行做了程序员。

2005 年开始在 Corezon 公司从事开发工作……做了很多 Java 电子商务项目，再后来，利用业余时间和朋友一起做了 DWZ 富客户端框架（JUI）。

## 门槛在哪里

在我看来，成为一个程序员的门槛几乎为零！

最大的门槛，就是自己！

如果你足够自信、自知，能够判断出来自己对这个行当感兴趣，那么一切都不是问题，不会的东西可以学习，互联网上的资源那么丰富，线下的培训班大门敞开，各类书籍汗牛充栋……试问有什么能阻挡你追寻的脚步呢？

**You Want, You Can, You Can, You Up!**

## 漫谈选人与培训

最近终于有时间来翻看《首先，打破一切常规》( *First, Break All the Rules* ) 了，感慨良多，真后悔之前公司发这本书时把它封印起来。

先看看书中提到的优秀经理们的座右铭吧：

*人是不会改变的。*

*不要为填补空缺而枉费心机。*

*而应多多发挥现有优势。*

*做到这一点已经不容易了。*

很简单的道理，我们都忽略了。在实际工作中，领导们、经理们、高级工程师们、研发人员，大部分人都有试图改变别人的实践、倾向。我们试图让一个人改变自己来适合某个职位，这是最大的错误。

对于选拔人员，或者简单地说招聘，其实也是类似的。在选拔人员的过程中，我们其实要识别的是：**他有什么样的才干。**

要做到这一点，就要区分技能、知识和才干这三个概念。之所以要区分，是因为我们习以为常地混淆了它们。

如果你曾经面试过别人，或者被别人面试过，就会有各种各样的体验。比如有的面试官就喜欢细细审视你的工作经验，用什么编程语言、什么工具，写了多少代码，负责什么模块……有的面试官天马行空，就问一些貌似不着边际的问题，比如你怎么向一个不认识的美女搭讪，怎样准备一次约会，如何与别人吵架并且要吵赢……还有的面试官会关注你玩什么游戏，是喜欢足球竞技，还是就爱魂斗罗……

其实他们都希望通过这些问题来判断当前的人选是否适合自己的需要。但更多的面试官通常分不清技能、知识和才干这几个概念。

技能，它所对应的是一件工作“怎样做”的问题。它们是人们可以相互传授的各种能力。对于程序猿，编程是一项技能，如果一名新手不会 Qt Creator，别人可以教会他啊。如果你不会用 Visio，别人也可以教会他。传授技能的最好办法是把整个操作过程分成各个步骤，让学生最后连成一体。

知识就是“你知道的东西”。比如你知道 C++11 的标准，比如你知道广电 181 号文，比如你知道你做的产品的卖点和要解决的用户痛点……这些都是可以教会的。C++ 语法、JavaScript 语法，这些都是知识，都可以教会。还有一些经验性知识则略有不同，它是无形的，只能通过自己回顾、归纳、总结来获得特定的感悟。比如有的人擅长讲 PPT，知道什么时候插入一些有意思的小故事、笑话来活跃气氛。

而才干呢，与技能、知识完全不同，它甚至可以说是与生俱来的。当你能够参加面试时，你的才干基本上都已经定型了。是不是很沮丧……难道我不可以培养某种才干吗？别担心，虽然你的才干在你十五岁时已经成型并很难改变，可是你多数时候并没有意识到它们的存在，只要你有意识地识别出它们并发扬光大，前景不可限量。

才干决定了你贯穿始终的思维、感觉和行为方式。比如有的人喜欢“精确”，这是一种才干，他可以成为优秀的会计师。而有的人马虎，你告诉他多次不要马虎了也没有用，因为他天生没有“严谨”的才干。比如有的人争强好胜，凡事都要做第一，不允许自己比别人差，只要给个对手他就能振奋起来。有的人天生热情，具有交往才干，热衷于结识陌生人，并且能够很快与别人打成一片；而有的人则羞涩，最怕与人说话，那你就不要企图让他变成社交明星。有的人喜欢有序，最不能容忍的就是杂乱无章，再乱的房间，她进去十分钟就会大变样，这也是才干，她可以做一个很好的物管、文员或者图书管理员。有的人就是会学习、爱学习，学什么东西都比别人快三步，不管是高数、微积分还是拳皇，这也是一种才干；与之相对的是有些同学怎么学东西都学不会、学不快，那是没这方面才干，也无法强求。如果你为人父母，就要识别孩子的才干并创造环境让他（她）发挥自己的才干，路越走越宽；如果非要拧着他（她）做不擅长也不可能变得擅长的事儿，众生皆苦。

任何可以产生效益的“贯穿始终”的行为模式都是才干。而取得出色成绩的关键在于使你的才干与工作相匹配。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

回到选拔人才这个话题，如果我们能够区分技能、知识、才干，那就能明白，为什么梁山需要鼓上蚤，而十一罗汉需要破坏天才巴世尔。你以为林冲、杨志可以变成时迁吗？不能。所以既往的经验不是最重要的，最重要的是这些经验里体现出他具有什么才干，而这种才干是不是你空缺的职位所需要的。比如你要招会计，面前的人连纽扣都系错位置了，我想他多半不是你需要的那个人。

当我从技术支持转做软件开发时，没有任何编程经验，C 语言也没学过，可是老板就收了我，后来也做得很不错。老板看中我什么？思维、快速学习与积极上进的才干。我带过的一个团队，有一个哥们就具有积极奋斗的才干，属于我们常说的自驱型人才，只要你给他活、给他资源，别的就不用管了，他总是按时或者提前完成任务，而且完成得很好，代码质量高，软件 Bug 少，连测试组的姐妹们都喜欢他。这样的人才不是通过培训就可以得到的。

所以，最关键的是如何找具有合适才干的人。然后就是给他环境，给他资源，让他最大限度地发挥自己的才干。

而培训呢？

培训到底能做什么？其实就是教会人们一些具体的技能和知识。不要指望可以把一个得过且过的人培养成极富责任感的业务尖兵，也不要指望培训可以改变一个人的习性。你能想象把一个见陌生人就低头、说句话就脸红的女士培训成左右逢源的交际花吗？有人不会用 redmine，不会用 jira，不会用 svn，可以培训一下，可是你能把一个凡事推诿不愿担当的朋友培训成项目经理吗？

才干是最重要的，却也是无法教会的。你教不会有主见，教不会善解人意，教不会体谅，教不会变压力为动力，也教不会毅力与坚持，更教不会百折不挠……市面上的各种管理类书籍千千万，各种培训机构团团转，它们都教你什么？如果你是想学某一项技能、某一类知识，没问题，它们可以胜任。如果你想学的是你不具备的某种才干，那就是缘木求鱼了。

所以作为单位，既要识别才干选对人，创造让人发挥才干的环境，又要明白该做什么样的有针对性的培训，将技能、知识与才干结合起来，产生最大的效益。而作为个人，虽然很难培养新的才干，但可以有意识地识别自己的才干，把你想要的若干种才干有机地联系起来，去做最能发挥你才干的工作，这样才能有所成就。

## 自我发现，找到适合自己的职位



“漫谈选人与培训”是我在读《首先，打破一切常规》时的小感慨，谈公司招人那一面的事儿。硬币的另一面，是作为公司员工的我们，如何找到一个靠谱的位置。这个很难说，因为每个人找工作的侧重点都不一样，我在这里也不打算对如何找到合适的公司、合适的职位来废话，而是想谈谈这个：**怎样判断你当前的职位是否适合自己**。话又说回来了，如果你能判断出当前的职位是否适合自己，就会对怎样找到合适的职位有莫大的帮助。

2014年国庆节带娃回了她姥姥家，7号回到西安，第二天早上送她去幼儿园，她不大想去，我和她妈妈就老生常谈，告诉她这种感觉怎么怎么样，反正就是想让她克服一下自己的感受，快到门口时她问我去上班是什么感觉，会不会不想去。

过完周末，第一天去上班，是什么感觉？

这就是关键之处，也是我们判断一个职位是否适合自己的起点。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

如果你想了解你正在干的工作是否适合自己，我建议你每个星期天晚上都给自己一点时间，梳理一下心情，搞明白“明天要上班”这件事儿给你带来什么感觉。

如果你内查到犹豫、恐慌、紧张、担忧、抗拒之类的情绪，停下来，再给自己多一点点时间，问问为什么会这样？工作中的什么让你产生这些情绪？——记录下来，列在本子上，看看哪些是临时性的、很容易消除的，找出那些较为持久的、难以消除的、让你感觉特别扭的，重点标记、跟踪。每周都这么问自己，每周都筛选让你不舒服、不愿意赶赴工作现场的那些因素。经过几周或者几个月，相信你一定会找到一些从你的角度看来坚硬如铁无法消融的原因，它们带给你压抑、抑郁，这些负面情绪大过你在工作中获取到的乐趣，以至于最终你过完周末就头疼要上班的事儿。

如果你发现自己非常渴望下一周的工作，有种恨不能穿越时空要去爱的感觉，停下来，快停下来，问自己为什么。当前工作中的哪些因素让你如此兴奋、享受、快感不断……找到它，记录下来，分析它们。周而复始，不断探寻、记录、分析……当你的周末自省结果发生变化时，看看是不是之前记录的结果也发生了变化。这样当你寻找下一份工作时，这些记录就能帮助你判断新的工作是否适合你。

我们说了两个极端，其实还有中庸的状态：无所谓抗拒与欢欣，就那样，啥也探查不出来，或者也懒得探查，爱谁谁。好吧，如果你连自己的感觉都不愿意关注，那也真是醉了……

不管你在内视时感觉到抑郁、痛楚、抗拒，还是欢欣、渴望、期待，抑或百味杂陈，都要忠实地记录下来，分析它们，区分哪些可以改变，哪些很难改变。

对于可以改变的，尝试去改变，因为它们一般都不是和你的天性对着来的。既然改变一点就可以让自己高兴一些，那就赶快行动吧。

对于很难改变的，要区分是外部环境（工作的物理环境、领导的行事风格、工作氛围、职位对某些技能的要求等）还是自身特性。如果是自身习性，那就算了吧，江山易改秉性难移，看来你的气场与这份工作不合，换个环境比较好，但是一定要明白，你的哪一点与这份工作中的哪一点不可调和，避免下次找工作踏进同一个坑里。如果是外部环境，其实可以尝试去改变，比如你不喜欢你的领导，可以申请调岗，比如你讨厌没有阳光的工位，可以找人商量换位置……

自我发现，找到适合自己的职位🦋

其实这种“探寻工作给自己带来的感觉”的内视过程，就是自我发现的过程。

有些人需要这样的显式过程，就像基督徒的礼拜一样，形式会强迫你的肉身和精神进入到一个平常难以抵达的境界。为什么武侠小说中很多人要闭关修炼，大概也是这个道理。

而有些人日三省乎己，已达随时随地感知自我的境界，则不需要这种显式过程了。这些人无时无刻不在修行，又何必拘泥于形式哉？话说这样的人也不会来看我讲这些，不说也罢。

自我发现的要点在于扬长避短，所谓尺有所短寸有所长，我们要慢慢地建立自知之明，了解自己的长处和短处，发现自己的独特才干，最终为自己的才干找到用武之地。

机遇只青睐有准备的人，快点扪心自问吧！



你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 亲爱的“程序猿”们怎么找工作



2002 年，我大学毕业，找了份售后技术支持的工作。2003 年的时候，奔波于河北维护程控交换设备。

我的第一份工作，是技术支持。干了两年多，觉得没意思，自己快被荒废了，于是决定转行做软件开发。先裸辞，后学了一星期 C 语言，幸好找到个公司肯接收我，谢天谢地，不然真不知道今夕何夕。那时被各种形势所逼，不得不换工作，所以虽然收入只有技术支持那份工作的 1/3，我也接受了。

一开始我在西安做 PC 软件开发，后来离开西安，去了南京，搞嵌入式，做导航、MP4、车载电子，干了一年多，又不干了。裸辞，回到西安。

做什么工作呢？其实我这人比较随性，再度回到西安时，已是 2007 年年底，工作近六年了，也没正儿八经想过这些问题。当年大学要毕业时，就更是少不更事了，人憨憨的，没见过世面也没什么想法，觉得有个事情干不错了。可人不能总这样，更何况我干了六年革

命工作了。

后来我找了一家做 WIFI 芯片的外企，待遇不错，当时给我将近八千元的月薪，公积金也都挺高。后来找了家做外包的，待遇也还行。再再后来我又回到第一家搞开发的公司了，工资是谈过的几家公司里最低的。是的，最低的。

为什么呢？

是因为老板对我说要做产品，做一个视频相关的产品，具体是什么还不清楚。那时我虽然还有点儿糊里糊涂，听过很多道理，还没过好这一生，可毕竟已经不是懵懂少年，多少有了些自己的看法。我觉得做产品比较靠谱，比外包靠谱，比外企有更大的可能性，风水轮流转，也许就可以鲜衣怒马走天涯。

**“有产品做，就不要做外包；有独挡一面的机会，就不要去做外企或者大公司的螺丝钉。”**这是我当年的一个认识吧。

当然很多人与我见解不同，理解并支持你们！罗素说，参差多态乃幸福之本源。本来么，求同存异，不能视见解与己不合的人为异端。工作中其实也是一样的。

一转眼，到了 2014 年。从 2008 年到 2014 年，发生了很多变化，不仅老婆孩子都有了，白头发也有了。

我已经三十多岁了，对于开发人员来讲，这是非常尴尬的年龄。就算你是个一线经理，其实也还是个干活儿的。

有次和朋友吃饭聊天，说起我们这个年龄的技术人员的出路，不免相对喟叹。如果要找工作，其实挺尴尬的，继续做开发吗？已经做了几年技术管理，不想玩儿过山车了。可是管理工作本身很难做不说，机会也不是太多，这时就体现出差别来了：招管理岗位的，更愿意接收有大公司管理经验的人员。而项目经理之类的，多数是外包公司或者定制开发类的项目，与做产品的初衷不同。

话说到这里，想必你也猜到了，我做了六年的产品，产品没有大红大紫……

其实干什么都是有风险的，对于开发人员来讲，最大的风险是什么呢？是选错一门技术，干错一个项目，做错一个产品，还是跟错一个老板？也许都不是。

## ☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

对于开发人员来讲，最大的风险是：**在职业规划上没有延续性地乱跳槽。**

有人跳槽为了涨工资，一年都能换几次，也不管工作内容、技术路线什么的。这其实也无可厚非。

我是一个比较迟钝的人，到了三十多岁才第一次正式地审视“怎样找工作、找什么工作”这个问题，可能是年龄大了吧，觉得不能再玩从零开始的游戏了。有人早慧，可能大学没毕业就明白了这个道理。我有一同学，毕业时放着微软不去，坚定地找嵌入式的机会，真给他找着了，并且干得风生水起。

当年我在参加一家公司的面试时对面试官说：要在两到三年内做到不可替代，结果被拒绝了，单位都讨厌这个。可是作为开发人员，这是你时刻要牢记在心的：**培养自己的稀缺性。**

为什么这么说呢？物以稀为贵。如果你干的活儿很重要，别人又替代不了，你自然会受重视。

如果你要寻找新的工作，此时一定要想：为什么跳槽？

是因为所用技术没有前景吗？比方说公司用 DOS 环境编程，或者一直使用 MFC，这都是过时的技术，如果不喜欢，就早日找个与互联网或者移动相关的方向吧，或者行业性很强又产能不过剩将来有大发展的方向，比如机器人、自动化控制、智能医疗、在线教育、智慧城市……

是因为工作不愉快吗？这个就要仔细掂量了，任何工作、任何公司都有让人不愉快的地方，人最大的敌人是自己。因此一定要想明白，自己的心态或者策略如何改变，从而才能改变工作状态。凤凰涅槃，浴火重生。简单说，改变自己，改变世界。当然，如果无论怎么样，某种不愉快都无法忍受，那就早点走吧。

是因为个人没有发展前景了吗？比如职位上不可能再晋升，技术上不可能再进步，薪水上不可能再大幅增加等等，这个就要想清楚了。

是行业突然遭遇暴风雨，前途黯淡了吗？如果是夕阳产业的话，那就没必要留恋了。

是公司突然遭遇资本危机，一蹶难再振吗？这就要看你自己的判断了。

是因为薪资太低，同行业为相比较，缺乏竞争力，或者简单说，生活压力大，入不敷出吗？这没什么好讨论的，因为这个换工作很容易被理解。

总之，要换工作，一定要想明白为什么。面试时这也是经常被问到的问题。

决定换工作了，就要考虑找个什么样的新工作。此时你之前的思考就派上用场了。机会只青睐有准备的人。

假如你已经很清楚自己将来的道路，非常笃定，坚信不疑，那就好办了，一定要找到这样的机会，抓住它。此时你的稀缺性，无论是技术上的，还是思想上的，都可以帮助你找到好的工作。而你对职业发展连贯性的原则与思考，也会让你的焦点不那么散乱，能够做到有的放矢不盲目。

最后说一下我对好产品的理解，大概是这样的：

1. 解决用户痛点。
2. 所处行业前景光明。
3. 产品有延续性，可持续性发展。

好产品加上靠谱的公司，基本上就是比较靠谱的工作了，甚至可以做成事业。

怎么判断公司是否靠谱呢？我有一些经验，用来鉴别不太靠谱的公司或工作：

1. 只关注你过往能力和项目的匹配度。
2. 只想用你已有的技术能力解决公司遇到的问题。
3. 面试你的人对公司产品没有热情，自己都不了解公司的产品远景，也不了解公司的远景。
4. 没有新产品，招聘是为了维护性开发。

如果新公司靠谱，有实力把这个产品做好，那最好。

那么怎样了解一个公司呢？通过互联网搜索、熟人打听、公司网站、招聘要求等，都是途径。还有一个途径就是面试。一定要牢记，面试是双向的，不单是公司选择你拷问你，也

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

是你最好地了解这个公司的机会，有时机就问一些思考过的问题，这也是前面提到让你想清楚为什么自己要离开现在的公司、自己想干什么、将来的路怎样规划的一个原因。我在面试一些应聘者时，通常在最后会留机会让他们提问，经常遇到有些求职者说我没什么问题，这么好的机会都放过了，一般这种情况下，这个人我就不再考虑了。

说这么多，其实是想表达，开发人员，也是要不断挑战自己的，那种“今朝有酒今朝醉，明日愁来明日愁”的做法是要被摒弃的，所谓“人无远虑，必有近忧”。在这个瞬息万变的IT江湖里，如果你不能识别并不断加强自己的竞争力，任由时间悄悄流逝而自己始终滞留原地，想必过不了几年，就该感叹了。

写给已过而立岁的自己，也给看本书的你：工作不仅仅是谋生的手段，也是人生的意义所在，无论有多少原因让你决定换份工作，但请谨记，只有让你感到充实、满足、有归属感的，才是你应该追求的，因为，它将让你成为你自己。

## 找工作的辟邪剑谱



我原来面试过一个哥们儿，半年内换了七家公司，我表示膜拜。

还有一些哥们儿，七年不换工作。这得是多稳定的工作啊，七年之痒都扛过去了，真心不错。

这都是极端情形，一般的程序员，可能会一年半载或者三两年换一次工作。在一个公司工作很多年的程序员，不是公司离不开，就是哪里也去不了。或许我说得太极端了，可能还有到哪儿都无所谓，能待着就待着的这种吧。

这次我们要说的是找工作，没错儿，就是找工作。

### 面试时如何展现自己

其实我这人比较懵懂，一直也没太搞明白如何展现自己的优点，甚至自己有什么优点也不甚明了。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

有人问，我怎么知道公司希望我怎么样呢？

其实要你自己做功课的，有几点是必需的：

- 仔细研究职位要求和工作内容，梳理自己与这些要求吻合的那部分能力、经历、知识。
- 仔细研究公司的产品或者项目，了解公司所在行业的背景知识、市场情况、公司的竞争对手的情况。
- 揣摩这家公司招你过去到底想解决什么问题。

## 目标

对于程序员来讲，也是一样的，我们也是人，也是有目标的。

生活的目标是什么？上帝把我们抛到这个尘世上来，除了让我们努力活得好一些之外，真不知道他还赋予了我们什么终极意义。

生活的目标决定了你工作的目标，所以，搞明白这两点其实还是有好处的。

你追求什么，决定了你要找什么样的工作以及什么样的公司。

几个月前我离职时，我的老板对我说：到了这个年龄，你缺的不是一份工作。

我深以为然：工作好找，找到能解决你为什么工作的工作难找。

## 简历

当你开始更新简历时，大概有两种情况：一种是想好了要去做什么，另一种是眼下的事儿实在干不下去了。换句话说，一个是主动去追求，另一个是被逼无奈要换个环境。

不管哪一种，要撰写简历是免不了的。

我换过七次工作，写过很多次简历。我被别人面试过不下二十次。我看过的简历超过三百份，面试过的程序员不下一百个。我想我可以从我的角度来谈谈简历里有什么需要注意的事。

- **为每一个公司的每一个岗位准备一份简历。**针对岗位要求来修改你的简历，别怕麻烦，一份简历通吃天下的做法只能说明你不勤奋，也不认真，那种把职位空出来，到哪个公司应聘就填写哪个职位的哥们儿，我看到直接就拒了……
- **不要超过两页。**我看简历一般不超过两分钟。有的人更短，只用 30 秒不到就下决定了。
- **教育经历要写明。**很多公司会用学校和学历筛掉一大批应聘者，因为他们相信，如果你能学能干，考上好学校不是问题，这可能有点儿以偏概全。反之，如果你来自一个名不见经传的学校而又毫不避讳地写出来，说明你有自信。
- **与对方要求相关的技能突出描述，放在所有工作经历之前。**很多公司招人时打的算盘是“招之即来，来则能战，战则能胜”，你如果不突出你的相关经验，八成没戏。
- **最近的工作在前。**过于久远的工作经历，提一下即可。
- **项目里说明自己的角色、用到的技术、负责的模块、完成的结果。**如果有攻坚克难的经历，要表述出来。
- **如果不是漂亮女生，就不要贴照片。**

一份好的简历会带给你笔试或面试的机会，值得你下功夫去琢磨如何在简历中展现你独特的一面。

## 关于内部推荐

其实还有一些其他的途径可以让你顺利闯过简历筛选这一关：内部推荐。很多公司推崇这个，认为自己的员工不会推荐不靠谱的人。我工作过的公司，如果你推荐的人被录用并且成功通过试用期，你会获得一千大洋的伯乐奖金。我爱人的公司也有类似的机制。所以，如果你能找一个目标公司的员工来推荐你，那你就踏出了成功的一步。

内部推荐之所以成功率高，是因为简历、笔试、面试这样一个流程，公司和应聘者之间还是不能够真的相互了解，这是因为语言是一种障碍，你说很多话，想要沟通，其实却会阻碍沟通。因为应聘者会不自觉地美化自己，公司也会美化自己。试用期虽然双方自由，但对大家都是巨大的成本。而内部推荐，它潜在的逻辑是：推荐求职者的员工应当对求职者有所了解，同时也对公司的职位要求有所了解，因此推荐的是相对适合的人；反过来，求职者也会通过推荐者了解公司



## 你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

的情况，消除求职者的盲目性。最终双方能够合拍的概率会大很多。

当然你不是每次都能找到一个人来推荐你，因为有的公司没有内部推荐这一说，也有的公司里没有你认识的人。但如果可能，还是要尝试这种途径。

六度空间理论（又名六度分隔理论）说：你至多只要通过六个人就能认识全世界的任意一个人。所以，理论上你可以在任何一个公司找到可以推荐你的人，只看你愿不愿意花心思。不要担心别人不愿意推荐你，多数人在帮助别人时是感到快乐的，不是有个词儿叫作“助人为乐”吗？

### 笔试



有的公司喜欢笔试，通不过笔试你决不可能成为他们的一员。

还有的公司喜欢出一些自己的顶级程序员也答不上来的古怪难题（比如计算某句谣言对某个人造成的心理阴影面积之类），他们的一大目的（也是部分老鸟程序员的乐趣）就是看着你出一些洋相，让你受一下挫折，打击一下你嚣张的气焰（想想那些监狱题材的影片里老犯

人对新服刑的雏儿的折磨你就明白了)，然后他们好自我优越一把。也许是我太邪恶，不惮以最坏的恶意来揣测别人吧。

当然也有的公司不笔试，只面试。

还有，有些公司的笔试是针对某一级别的求职者。也许你现在到哪里都不用笔试。

如果你必须笔试，我没什么好说的，只有一点，你还是要研究你所应聘的职位有什么要求，然后揣摩笔试的范围，这多少会有一些用，比如他找懂 C++ 的程序员，自然不会考你 Java。

市面上也有很多类似《C++ 笔试宝典》之类的书，会总结各类公司常出的笔试题并给出解法，Java 的也有。

## 面试

笔试之后就是面试了，也许你一个小时后就会接到面试邀约，也许你等到花儿都谢了也没有音讯，反正大多数公司只会通知通过的那些求职者，对于被涮掉的，潜规则是无视。所以，这一切其实不那么美好，也许你觉得自己题目答得不错，没 100 分也有 99，那你就是抓破头皮也想不明白为什么不通知你面试了——其实有时候答得太好了是一种错，他们可能怀疑你利用智能手机在线求助，或者认为你熟读各种笔试宝典，或者批阅试卷的人会认为你太优秀将来会对他们不利……总之各种可能，也可能是 HR 或者前台觉得——回复应聘者过于麻烦。

程序员有一个问题，就是沟通障碍。这是双方的，是面试官和求职者都存在的问题——“一面”通常都是来自技术线的优秀程序员。如果你和面试官投缘，一见钟情，那恭喜你，你一张嘴他就知道你想说什么，你 99% 可以通过面试。如果你和面试官不登对，一见面就气场不合，大眼瞪小眼，表面彬彬有礼，暗地里已经刀枪剑戟了，那你 99% 会被淘汰。更多的时候，你和面试官对方都没什么感觉，既不来电也不反感，你们需要用语言来沟通，在一问一答中表现自己。

因为面试者和求职者地位上的不平等，会对面试过程产生一些微妙的影响。这里建议你站在公司的角度来考虑一下，其实他们也是有求于人，并不是你热脸要去贴冷屁股。多数公

## ☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

司愿意走面试环节，都是因为真的需要人，不然劳民伤财为哪般。如果你想明白这一点，就无须紧张，也无须仰视那些让人膜拜的面试官，你可以放平心态，侃侃而谈。如果你不善言辞，也可以自然地展现你的实力，就用你以前工作的方式来展现自己。

很多自认为能力不错的程序员往往在面试时折戟沉沙，我也是这样。这里面的一大原因是我们不知道如何展现自己的能力，是滔滔不绝雄辩天下，还是一问一答诚实木讷。这是大问题，语言本身就是迷雾，会给沟通带来很多障碍，你心里想的是这样，说出来就带了起码20%的误差，而听的人又有自己的过滤器，他们只会听出自己想听的那部分而忽略你想说的那部分，所以要想彼此理解，是天大的困难。“听话听音”这个词儿就是描述使用语言沟通的困境的。

那么语言究竟能不能表达出程序员的能力？答案是不能！

因为软件开发的特性，程序员做得多说得少，久而久之更善于用简单直接的逻辑来表达想法，不善于琢磨话语背后的意思。而面试却恰恰需要琢磨话外音：到底面试官问这个问题的意图是什么，他希望考察应聘者哪一方面的技能和见解，他某一句话对应聘者是肯定、否定还是无所谓，应聘者还要不要继续展开来说。等等，所有这些不确定，会让求职者产生焦虑和惶惑，原本可以表现十分的，只能表现六分。当然也有一些人语言能力很强，眼光也很敏锐，脑子又比较灵，兼之略懂“读心术”，能够看透面试官转笔筒或者跷二郎腿背后的意思，也能够理解面试官语气的变化代表的心思转折。这样的人是人才，可以通过大部分的面试。

生活在“零壹”世界里，我很容易采用二分法。其实程序员有两种，一种是茶壶里煮饺子——有才说不出；另一种是说的比做的多，一分的才可以展现出十分来。到底哪种好，看站在谁的角度，反正这两种类型的人大多相互看不起。而对于公司来讲，其实两种人都需要。

既然对话不能了解一个程序员真正的实力，那为什么还要走这一套程序呢？答案是：没有更可行的方案。

现代人在生活中已经比较开放了，在招聘与求职这么重要的事情上，却保守得很，不肯采用更合适的方式方法，比如让一个程序员到公司工作几天试试双方是否合适。其实这种方

式好处非常多，因为只有近距离接触才能使人看得更细致，避免那种远距离的看上去很美带来的悲剧。

那么究竟面试时程序员该如何展现自己呢？

- 知道你擅长什么。
  - 知道公司需要什么。
  - 不要争辩、争执，哪怕对方误解了你。
  - 知道面试官的身份，比如是 HR，是程序员，是项目经理，部门经理，还是技术副总，这很重要，不同角色不同层次的人，听话的方式是不一样的，你需要根据这个来调整你说话的方式，以便对方可以听进去。下面就是 switch-case 语句了：
- ✧ 程序员：把他当作你的同事，以讨论技术问题的方式交流，表现出你解决问题的能力。
  - ✧ 一线技术管理者：多听，理解问题，知之为知之不知为不知，多数时候不要表现得比他们还聪明；表现出你团队协作的一面。
  - ✧ 技术高层：表现出你自己的想法、看法，表现出你对公司技术方向、产品方向的了解与见解，当然也要表现你的技术能力，无须担心你太强吓住人家。
  - ✧ HR：友善地和他们谈话，放下你程序员技术高于一切、天下唯我独尊的想法，说人话别说 IT 黑话，别看不起不懂技术的人。

面试可能有好几轮，有时一天就结束了，当场就拿到 Offer。有时会拖几个月甚至半年以上。此时你要有足够的耐心，相信好事多磨，要是等不了，也没别的办法，可以签别的公司。

## 为什么我会被 PASS 掉

有时还没到谈薪水的时候消息就石沉大海了……

为什么？

你很想知道答案，可是没人告诉你，你只能瞎琢磨。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

其实这是求职者和公司之间地位不对等的问题，公司觉得拒绝一个人可以不需要理由，但对于求职者，如果能知道被拒的理由，会是非常好的一件事情。

有几个常见的原因：

- 面试官觉得你能力不行
- 面试官觉得你能力太强
- 面试官觉得你不稳定
- 面试官觉得你个性太强
- 面试官听不懂你说什么
- 面试官觉得你听不懂他在说什么
- 你提的工资要求太高，超过 HR 拿到的上线太多

## 骑驴找马

裸辞是需要勇气的，几个月的空档期很多人承受不了。所以，相当大一部分人是找着了下家才离职。那么，一边工作一边找下家，究竟有什么影响呢？

有个词儿叫“心猿意马”，还有个俗语叫“身在曹营心在汉”，用在这种情况下都很合适。



当你决定找工作时，对现在的工作就会产生懈怠，这是毋庸置疑的。别说你多有职业精神，那都是空话，懈怠是必然的。即便你努力表现得和平常一样，但是一旦你想到不久就要离开，对很多事情就没那么尽心了：可能会把有些事儿推给别的同事；可能不再愿意为工作而加班；可能对手上的问题睁一只眼闭一只眼；可能会下意识地让自己空忙；可能难以集中心思来解决复杂的问题；可能对 **BOSS** 也没那么在意了……凡此种种，都会有外在的表现，发出各种信号，明眼人会看出来你已经动摇，心思不定。所以说，当你决定骑驴找马时，效率会下降，对公司的产出肯定会减少。

那么对自己呢？其实也是一种煎熬。人都有站队的思维定势，希望在某一个集体中获取认同，不想像蝙蝠那样长久骑在墙上。你会担心现在公司的人看出你的马脚，会尽力掩饰蛛丝马迹，这会让你很累。你还会因为迟迟没找到马而焦虑：已经决定要走了，却找不到下家，这种情况会让你对自己的能力产生怀疑，让你有时沮丧有时疯狂有时黯然神伤。有时你还会觉得对不住现在的公司，你内心中有两个小人儿会经常打架搞得你疲惫不堪。当然也有强人，对这一切都不在乎，因为他认为到更好的公司获取更好的职位是天经地义的，绝不会像我们这种小工蜂一样拿忠字舞来束缚自己。

不管怎么样，骑驴找马是一种常见的存在，你、我、他，可能每个人都曾经这样做，也相信别人会这么做，**BOSS** 们也常为此焦虑，但是，该发生的发生了，不该发生的也发生了，我唯一的建议是，无论怎样，还是要站好最后一班岗。

你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 薪资，你是我不能言说的伤



这次我是想谈薪资的问题，话题有点儿敏感了，不过大家私底下已经背着公司“禁止打听同事薪水”这种规定聊过很多次了，现在我们台面上说一说也无可厚非。

### 钱、工作、程序员

我们不谈钱，谈钱太高雅了，俗人都怕谈钱。我们只说几种现象。

如果把程序员的世界粗暴地按钱、事二维化，程序员可以分为下面几类：

- 拿多少钱干多少事儿，工作一分不多干。
- “begin:” 先干事儿，再谈钱，事儿成啦，钱也得给我涨上去“goto begin;”。
- 干事儿，一直干，不谈钱，相信公司会给我应有的回报，短时间的委屈也没关系。
- 先谈钱，再干事儿，钱上受委屈了，什么事儿也别想让我干，钱到位了活儿才能干得漂亮。

薪资，你是我不能言说的伤🐼

- 钱够了我也不好好干，怎么着吧，那谁谁拿得比我还多呢。

我是写软件的，总想覆盖得全面一些，逻辑上不要有漏洞，不过对于优秀的程序员来讲，“想得多，写得少”也是一种追求。

## 面试谈薪资



有很多朋友认为起点很重要，会去研究各种策略。

要是问我，我只有一点建议：先问明白公司让你做什么，你能帮公司带来什么价值，然后再谈钱吧。

我还可以说说我遇到的几种 HR：

- 就这个价，爱来不来，多一分都不可能。
- 想尽办法打探你的底线，然后给的钱比你的底线少一些，能少就少。
- 友善地告诉你公司的底线（实际上已经打过折扣，距离你期望不太远），然后让你考虑。
- 谈公司未来的发展，始终不动声色，不暴露底线。



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

其实我觉得，这都无所谓，如果你为自己定了底线，那就盯着它谈，谈不拢也罢。

如果你很喜欢这个事儿，很在乎这次机会，那另当别论。

一千个人一万种选择，但有两点最重要：

- 知道自己的价值是否符合对方的需求。
- 对方要你做的事儿你是否喜欢。

## 实战面试谈薪水

在面试时和公司谈薪资，有几点是要注意的：

- 不要在“一面”、“二面”、“三面”等过程中主动打探薪水、福利等，一般这会给对方留下比较差的印象，认为你只在乎钱。
- 在确认公司准备聘用你之前不要谈薪水，因为只有公司决定聘用你的时候，才是合适的时机。
- 确定谁是要和你谈薪水的那个人，有时是 HR，有时是副总、总监之类，你无需询问，也会有机会知道。如果是 HR 来和你谈薪水，那他们多半是从技术线拿到了一个范围，然后会先压低了薪水来和你谈，要做好心理准备，适当坚持一个更高的薪水。如果是技术线上的领导直接和你谈，我想你先要对面试的过程有个总结，了解这个人对你的印象，他是对你有强烈的兴趣，还是觉得你差强人意。如果是后者，你坚持要获得更高的薪水，很可能没有结果。但无论哪种，你都要对这个职位表示出足够的兴趣来，同时也要表明你在意的不是薪水本身，而是薪水是否能与你的能力以及你可以给公司带来的价值相匹配。
- 被问及期望的薪水这类问题时，不要说具体的数字，如果一定要说，说一个范围。
- 不要透露你现在的薪水，因为公司往往会根据你现在的薪水来给你定价。
- 如果实在躲不过，一定要说现在的薪水，切忌撒圆不了的谎，虚报一大截子并不是明智的做法。HR 们远比你想象的有能力，他们很可能和你之前所在公司的 HR 或其他人认识，能在做背景调查时问到你的薪水。
- 如果对方拿你现在的薪水说事儿（碰巧你现在薪水较低），你应该表明你现在的薪

薪资，你是我不能言说的伤。🐼

水与能力严重错位，是不合理的，体现不了你的价值。

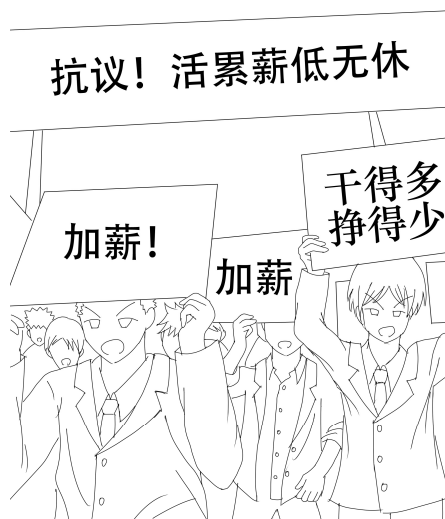
说到底我也只是泛泛地遐想，给不出什么秘诀。

不管你了解多少谈判技巧，摆平自己的心态很重要。有一部分程序员觉得谈薪资谈来谈去伤感情，其实不必，就跟你在商场买衣服一样，争取到合理的价位是人之常情，而且如果你不情不愿地进了公司，事后发现被压低了工资，不管对自己还是对公司都会产生非常消极的影响，所以呢，你肯进入公司之前谈个明白，是对双方负责任的体现，因此你不必担心这个过程会有多大的不良影响。

## 加薪在囧途

在“程序员的生活就这样吗”中我好像设计了一个谈加薪的对话。我要说的是，这都不是真的，却胜似真的。用一句话概括程序员的加薪之路，那就是：满纸荒唐言，一把辛酸泪。进入每年自动加薪 30%环境的兄弟姐妹们请一边儿安静地数钱去吧。

## 为什么你要加薪



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 生活压力

相当一部分程序员工作的唯一原因是薪水，谈理想、谈情怀、谈抱负，那都是忽悠人的。和雇佣兵一样，工作的目的就是获取薪水，获取更高的薪水。这是生活的基本问题，没什么不好意思呢。再说，都说程序员吃的是青春饭，前半辈子拼了身体赚钱，后半辈子拼了钱修理身体，如果前半辈子拼了身体却赚不到钱，后半辈子怎么办？

另外随着年龄增长，无老可啃的那些人，要面临严峻的生活压力：取媳妇、买房子、办婚礼、生小孩、抚养小孩、赡养老人……因此，路漫漫其修远，我们不能没有钱。

先要生存，而后才是凌空蹈虚的理想、抱负、情怀。这就是我们普通人的路子。有的人虽穷困潦倒却心系苍生，若千年如一日不改初心。这是我们要敬佩的，但不一定是我们能够做到的。“梦想还是要有的，万一实现了呢”这种说法当然很好，我也喜欢，只是很多时候我们都忘了怎么“做梦”了。

总而言之，当我们这些普通的程序员面临生活的压力时，就会要求加薪。

## 能力肯定

据说现在很多大学生毕业时零薪水就业，当然这是特例。

刚踏上软件开发之路的程序员各项技能都比较初级，当他们工作一段时间后，觉得自己能力提升，已非昔日吴下阿蒙，会自觉地产生提升薪资的想法，毕竟，薪资要与能力匹配，技术很强，薪水很低，这种落差一般人都不容易接受，除非怀有特殊目的（比如在某个公司工作不为钱只是为了那个她）。

## 水涨船高

程序员是一个很大的群体，每个程序员周围都有一批同行业的其他程序员，有时难免左看右看上看下看，原来每个阿猿薪水都不简单。于是，需求从比较而生：为什么那些哪儿都不如我的人拿得都比我多？不免产生明珠暗投、怀才不遇、物非所值的落差，此时就会不平衡，产生强烈的加薪欲望。

## 通货膨胀

钱印得越来越多，钱也越来越不值钱了。几分钱一斤的大白菜再也买不到了，一元一票制的公交车越来越难挤了，想当沙丁鱼罐头都不容易了。

国家五年规划里说五年内人民群众收入翻番，起码每年涨幅 15%才能不给国家丢脸。可是我在单位干了两年了还不涨工资，实在是对不起国家，对不起江东父老，要不是还没有找到虞姬，也得学楚霸王剑吻乌江了。

## 为什么不给你加薪

看个故事吧。

### 又一个阿猿加薪的故事

阿猿最近心塞得要死，天天晚上睁着眼睛数天花板上的蜘蛛网到底有多少个结，可惜数了一星期也没数过来，眼睛却爬满了红色的蛛网，白天只好带着熊猫牌眼镜上班，为怕同事和老板发现打瞌睡，还得学 TOM 那样搞两根不带尖儿的牙签来撑眼皮儿。这日子过得！

为了对抗疲累，阿猿抽的烟越来越多，从一天七八根发展到十七八根，又从十七八根发展到二十七八根，这速度，比薪水涨得快多了。每每在抽烟区碰见老板，老板都要问一句：“阿猿，最近抽烟很凶呀，是不是有什么烦心事儿？”阿猿只好说：“嘛事儿没有，就是上瘾了。”

除了抽烟，夜深人静独对电脑时，阿猿还大量饮用劣质黑咖啡等怪味饮料，以期激发自己体内潜能。他真是发了狠：我就是不信我这么拼老板还不给我涨工资！

疑心遇见鬼。怕什么来什么，说的就是阿猿了，果然，这次加薪又没有他。

阿猿痛苦地闭上眼睛，而双耳却情不自禁地收听周遭的信息，伤心、失望、愤懑、不甘，种种滋味萦绕心头。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 技术副总的说法

阿猿内心挣扎了很久，终于决定找技术副总 L 谈谈。

L：“阿猿啊，最近活干得不错，可是看你精神好像不太好，是不是压力太大了。”

阿猿：“我很拼的，一心就想把产品做好。”

L：“咱们就需要像你这样敢拼敢干的带头人。”

阿猿：“可是我心里很失望。”

L：“怎么了？”

阿猿：“L 总，你说我干活怎么样。”

L：“努力程度大家有目共睹啊。”

阿猿：“L 总，我这人直，说话不会绕弯子，我就想明白了，为什么我这么努力，这回涨工资却没有我？”

L 饶意味地看着阿猿，看了很久，没有说话。

阿猿心中千头万绪，前尘往事都上心头，一时之间忽觉万事成灰，争与不争又有何意义？可一颗心却不能平，委屈却不可抑制，终于，英雄虎泪冲出熊猫眼眶，若江河崩塌，汹涌之极。

L 从办公桌后面转出来，靠近阿猿，伸手用力拍拍他的肩膀，悠悠地叹息一声，静待阿猿平静下来。

良久……

阿猿：“多少年没掉过眼泪了，让 L 总笑话了。”

L：“你是性情中人，看得出来。”

阿猿：“L 总，不是我受不起委屈，实在是我付出了太多。”说着话阿猿伸手抚了抚腰，

薪资，你是我不能言说的伤🔪

又抬手撸了撸脖子梗，然后说：“L 总，你看看，腰椎间盘突出、颈椎强直，这都是我前段时间加班加出来的，我……”

说话间，阿猿再次泪崩。L 只好又从座位转出来，再次揽一揽阿猿的肩膀，嘴里说：“不管工作多忙，身体是最要紧的，你的努力我都看见了，大家也都看见了，放心，你的努力不会白费的。”

阿猿：L 总，我就想明白了，为什么不给我加薪？

L 反问：“你觉得呢？”

阿猿：“我就是想不通，L 总你帮我分析一下。”

L：“我只能根据经验说说，也都是推测，不见得准确，你听听就算了。”

阿猿：“L 总你只管说，我就是想知道为什么。”

L：“去年我们的产品做得怎么样？”

阿猿：“大家都很努力，天天加班。”

L：“那产品有没有问题？”

阿猿：“每个产品都难免会有问题。”

L：“从结果上，产品没有如期交付。延期交付之后还出了几次事故。老板的考虑应该是论功劳不论苦劳。”

阿猿：“我们真的很辛苦。”

L：“其实我从老板那里了解到，他很想给每个人都加薪。加薪了大家高兴，活也干得好。可是这两年经济实在是不景气，公司的产品老被客户投诉，不但没赢利，还要赔钱。老板也很难做。”

阿猿：“说好的半年加一次薪，我都一年半……”

L：“公司的规定是一年两次加薪的机会，也就是说，机会，可能有的人一年加两次，

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

有的人两年不加。”

阿猿：“L 总，我就是那个两年不加薪的人！这对我公平吗？我都累得一身病了！”

L：“下次一定会有的，其实这次的大名单里有你的，我可以给你看邮件，但最后因为种种原因没有给你加吧，我觉得，要受得住委屈，男人么，都是在委屈中长大的，你放心，今年年中见分晓，你这么拼，如果产品状况好转，绝对没有理由不给你加薪，到时你的加薪幅度肯定甩其他人一条街。”

阿猿：“L 总，有您这话我就放心了。”

L 望着阿猿远去的背影，心里默默叹息一声：老子也没加，年终奖还不知道有没有呢！

## 加薪的时机

有的公司每年两次调薪机会，就像上面的阿猿所在的公司，一次在年中，一次在年底。有的公司一年调一次。

不管怎么样，大部分公司都有固定的调薪周期。如果你有加薪的想法，在接近调薪周期时，提前一段时间（比如一个月、半个月），找领导讨论这个问题是比较合适的。

另外还有一些时候也是可以谈的，比方说某个产品或者项目成功发布，而你在其中发挥了重要作用，立下汗马功劳，那在庆功会或项目总结会议之后，可以找一个机会，不着痕迹地跟领导提一下。此时也有很大的成功概率。

还有，需要注意的是，如果公司业务上和资金上正遭遇困难，就不要在这个节骨眼上提加薪的事情，一般不会获得你想要的结果，甚至还可能让领导对你有异常的看法。

## 实战谈加薪

这又把我难住了。要知道，作为程序员，我本人很少找领导谈加薪的事情。我秉持公司不会亏待劳模的想法，相信酒香不怕巷子深，以为只要是金子，总有发光的时候。可实际上，就算你是锥子，也有扎不破麻袋难以出头的时候呢。

薪资，你是我不能言说的伤🔪

## 准备好你的理由

一定要提前做工作，准备好你认为足以说服领导的加薪理由。

比如，你的能力提高很大，以前只会 HTML 与 CSS，现在呢，JS 会了，Node.js 会了，Perl 会了，MySQL、Oracle 都会了，Java 会了，Python 会了，Docker 也熟练了……那么，你已经非常强悍了，快成为“屌丝战斗机”了，难道还不该加薪吗兄弟！

比如你在最近的项目中大展身手，承担了核心模块的工作，以难以置信的漂亮姿势完成了任务，拯救了陷入泥潭的团队，那也是非常强的加薪理由。

又如你已经两年没加薪了，而这两年里你做了 A、B、C、D……等足以谱写一首字母歌的项目，已经量变了，再加上同事们都涨了，就你被遗忘到了旧时光里。

……

凡此种种，一定要是相对客观的理由。

## 找谁谈加薪

要谈加薪一定要找对人，表错了情是很尴尬的。

有的公司是老板说了算，项目经理就是个带领大家拼命的角色，除此之外没有其他权力，那你和项目经理谈加薪基本上没什么用，他还要一级一级往上传递你的请求，很可能到最后就“走音”了。你要想办法了解这个链条中谁说了算，找准时机和他谈话。关于这一点可以找你的前辈们打听打听，或者和项目经理聊天时迂回地刺探一下。

有的公司是部门经理有决定谁加薪的权限，那你就找部门经理谈。

总之各种不一样，因环境而变。

## 如何谈加薪

很多程序员和电脑沟通比较顺畅，私底下和同僚沟通也比较顺畅，可以用妙语连珠来形



## 你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

容。可一旦和领导沟通，就比较拘谨，放不开，无法准确地表达自己的想法，这会给谈加薪这件事带来一些不利影响。不过没关系，精心准备会提高成功率。

### （1）当面比书面好

这是自然的，你可以看到领导的反馈，也可以声情并茂或者涕泪横流地表达你的愿望，比电子邮件里不痛不痒的几句话更容易出效果。

### （2）提前设计谈话路线

凡事预则立，不打无准备之仗。

当你找到了足以说服自己的理由也自信能够说服领导时，就要考虑如何面谈。比如怎么开场，怎么聊公司的业务发展，怎么聊自己的能力提升、贡献、生活压力等，总之要先暖场，描述现状，表达自己的理由，提出加薪的请求。

要清楚公司、产品、项目的现状，揣测领导可能以什么方式来回应你，你要想好应对策略。一般领导会比程序员更善于引导谈话的方向，而程序员能做的就是做好功课，拟定目标，温柔坚定，不要被受领导影响。

### （3）谈话的态度

不要太强硬，不要带威胁的意味，哪怕你真觉得自己功高盖世，无可替代，公司缺了你转不成，也不要威胁领导给你加薪。领导可能一时被你威胁，你可能成功一次，但第二次一定会出问题，他一定会培养他人来替代你。

如果公司效益不好或者眼下遇到困难，而你又觉得非加薪不可，那当领导以类似的理由来延迟加薪时，一定要作出聆听的姿态，表示理解公司的难处，耐心等待领导的讲话告一段落，然后再重复你的难处，温柔地表达你的决心，即便最后领导表示实在无法立即满足你的要求，也不能当面争执搞得下不来台，你只要表达希望领导考虑自己的请求就可以。至于你下来准备换东家什么的，那是接下来的事情。

### （4）一次足矣

找领导谈加薪，一次就够了，如果一而再再而三地谈，不但没用，还容易引起领导的反感，让领导觉得这人就只在意钱，以后对你的态度必然发生变化。

## 加薪的结果

加薪谈判的结果，要么失败，要么成功。

成功加薪了，亲爱的“阿猿”，肯定会以专业的态度努力工作，干出更好的成绩来。

加薪失败，这里就有个选择题给你做：

- 极度不满，跳槽。
- 不满，暂时又没别的打算，又担心领导因此对自己形成了不好印象，陷入犹疑不定之中

如果你选择跳槽，那就没得说喽。

如果你暂时不想跳槽，那就要让自己想办法走出加薪失败的阴影。请回过头来考虑公司的现状和产品的销售情况，再次梳理自己在大大小小项目中的表现，为下一次谈判做准备，千万不要气馁或者满腹怨气，陷入循环往复的烦恼当中，那样对谁都没好处。因为加薪这种事，真的是“尽人事听天命”，不是你能完全把控得了的，失败也不是什么大不了的事儿，继续提高能力做出更好的成绩才是正道。

你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 程序员该不该考虑初创公司

IT 这个行业，最不缺的就是机会，追逐机会的，有各种各样的小公司，也有各种各样的程序员。

### 创业公司

每一个出来创业的人，心中都充满了希望和梦想，而且认为自己眼光独到，手段高超，一定能够逐浪潮头。



可是，创业成功的概率到底有多大？十分之一，还是百分之一？

熙熙攘攘的创业大军，在互联网的江湖里闯荡，有成为传奇的，也有成为炮灰的。传奇者的传奇鼓舞着后来的创业者，创业者已经做好了准备，所谓艰难险阻，不过如此尔：



只要我们找到合适的切入点，找到合适的人来推动创业的轮子，总有一天会登上巅峰。

创业公司有个很大的困难：人才！

简单粗暴地划分一下，创业公司有两种：

(1) 有钱的。

(2) 没钱的。

很多创业公司是含着金钥匙出生的，创始人有显赫的经历和卓越的能力，很多 VC 和天使追着要投钱，也有很多程序员死乞白赖地要跟着他们闯荡江湖扬名立万。有钱、任性，怎么都能招来人。

也有很多创业公司，不那么有钱，他们的创始人认为自己抓住了一个想法，这个想法在他们看来独一无二，值得赌一把。人不轻狂枉少年。这些创业公司，有一点钱，有一点想法，但钱不太多，在启动后，很快就会面临人才的窘境：没有背景，没有追随者，没有钱，不能砸晕程序员。

## 程序员啊程序员

互联网江湖的创业公司，会需要大量的程序员来加入他们，一起实现梦想。可程序员是怎么想的呢？

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 追求稳妥

我本身是一个程序员，还是老程序员。我的孩子虽然还不会写程序，但也会打酱油了。像我这样的程序员，会有很多顾虑。

首先我们要养家，你知道，房子很贵，房贷很烦；孩子很可爱，花销太大。虽然我们可能每个月拿可观的薪水，可那都是账面上的，只是从我们银行卡上过一下，没多少可以称为财富的，因为转眼它们就没影儿了。

对于工薪阶层来讲，如何衡量你有多少财富？

有一个简单的算法，在不工作的情况下，你拥有的钱物，能让你在保持当前生活质量的前提下生活多少年？

我算了一下，对我而言也就是一年吧，顶多是两年。

所以，靠薪水过日子的程序员，像我这样的，其实都有一种无形的压力，不敢停下来，三天没工作就心慌，半个月没活儿干就坐不住了，一两个月没钱进账就要死要活了。

对于这样的程序员，受生活逼迫，找工作首先考虑的是稳定，面对找上门来的创业公司，他们的犹疑是必然的。

## 管你怎么样，我就拿老子那份钱

很多技术人员看问题简单直接，我们程序员更是和机器打交道惯了，形成了干多少活儿拿多少钱的习惯，别的什么事儿也不操心，操心有什么用，多操心你又不会多给我钱。技术人员还有一个普遍的想法：工作多得是，此处不留爷，自有留爷处。所以，相当一部分技术人员，其实并不是那么关心企业的死活。

别急，在你准备好棍子要揍我先想想是不是这样。

当然公司赚钱多，程序员也可能拿的钱多，可程序员觉得这是不确定的，钱都是老板的，赚再多和我有什么关系？你敢说自己真没这么想过吗？

所以，我就干我的活儿，拿我的钱，管你公司是赔是赚，该涨工资了你还得给我涨。这是相当一部分程序员的真实情况。

这样的兄弟姐妹们也可能会和创业公司碰头，可是创业公司不喜欢这样的程序员。

## 追梦人



有梦想的人很多，有的人有资源出来创业，还有的人心中燃烧着梦想但不一定自己创业，他们可能会选择加入一个创业团队，在创业平台上实现自己的梦想。对于这样的追梦人来讲，选择什么样的创业公司，就是他们要仔细考量的。

假如一个初创公司要做的事儿看起来很靠谱，成功的可能性很大（创业者是这么想的，别人可不一定），程序员在自己的角度来琢磨也觉得是这个样子，那两者一拍即合的可能性就很大。

如果这个初创公司再有资本，所给的薪水也有吸引力，还能搭配点期权之类的话，我估计这事儿就成了。

## 创业公司要甚样人

创业者说，从我们立起门户的那一刻起，我们就知道，成功的关键是：人，人，还是人。

你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

那他们要什么样的人？

队友。

创业公司在初期最需要的是队友，能够和公司一起前进，与公司共同进退的队友。虽然这样的人不大好找，但他们会成为核心团队成员，也可能会成为技术合伙人。

有些人创业，“立山头”那一刻都是带着队伍的，所以核心团队成员都已经就绪了。

有些人创业，是有个想法拉点钱找一搭帮的，他们的核心研发队伍都还没谱呢。

不管哪样的，他们都需要这样的人：

- 可能成为队友的人
- 愿意与企业一起成长的人
- 有想法的人
- 有激情的人
- 拒绝混日子的人

阿猿，你身边的朋友创业了吗？他们拉你了吗？

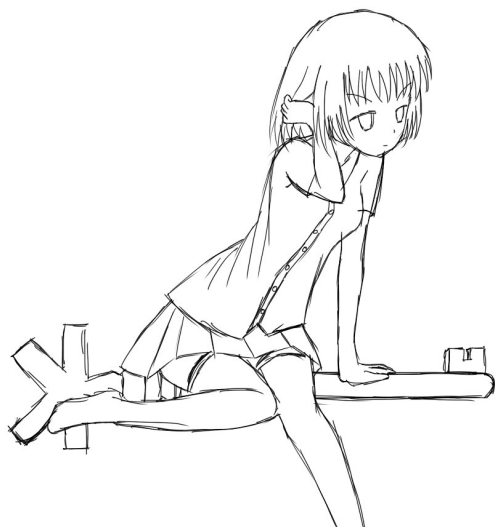
## 程序员与初创公司的约会

我们选择一个初创公司，总要图其中一样：  
方向、金钱、人。

当然你也可能考虑其他因素：地域、工作强度……但前面的，可能更重要。

### 方向

不论是创业者还是程序员，首先要考虑的就是创业方向。只有方向对了，才可能找到你的



金钥匙。

方向，对每一个人的发展都很重要。只有选择了一个合适的方向，沿着这个方向走下去，才会走得更远、走得更好。没有方向的路，只会让人慢慢迷失，永远也走不下去。可现实是，很多人都没有方向感，包括我自己，也常常是走哪儿算哪儿。

不过，我觉得过去没有方向不重要，但现在一定要开始想自己的定位和方向。如果想明白这个，以后就不会再有大的迷惘。而一旦你想明白了，那什么样的公司适合你，什么样的公司不适合，都是很容易判断的事情。

## 金钱

工作的一大目的就是获取薪水，获取更多的薪水，让生活更美好。这无可厚非。所以从这一点出发，用“钱景”来决定选择什么样的公司也是相当一部分程序员的出发点。

没有最多，只有更多，越跳越多……这是很多程序员的工作路径。

## 选人

大家都看过《西游记》吧？

有个问题：唐僧为什么是西天取经集团有限公司的老大？

唐僧武力不如悟空，圆滑不如八戒，扛包袱不如沙和尚，也没一样神奇变化，他为什么当老大？

最关键的就是他这个人，他就是个当老大的人，就是个创业者的角色。

大概来分析一下吧。

首先，唐僧长得好，白白净净，符合眼球经济的规律，不但女妖精要死要活地要得到他，男妖精也要活要死地要吃他的小鲜肉。

其次，唐僧有背景，人脉广。你看，他是金蝉子转世，金蝉子是如来佛祖的徒弟，这关系不错吧。还有，观音也老罩着他。这是天上。地下呢？他跟大唐皇帝也是铁哥们儿，皇帝



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

那是他的死忠粉儿，不但赠予锦斓袈裟，还洒泪送出三百里呢。

最后，除了人脉，唐僧还有一样法宝，方向感强，目标明确，坚毅不拔。他的目标就是取经，从来不曾动摇，温柔乡不能陷之，权力冢不能埋之，没马可以步行，没脚可以爬行……这样的人不当创始人谁当创始人！

程序员面临一个初创公司，很重要的一点就是，看他的创始人是不是靠谱，如果创始人靠谱，那就没问题。

假如你很难了解创始人，也要想办法了解公司的合伙人、员工，所谓物以类聚、人以群分，看一个人如之何，就看他的朋友如之何，就看他身边的人如之何，这样做没大差错。

**跟对人，干对事儿。**说起来很简单，可实际上，如何确认你要跟的人是对的，如何确认那个人要做的事儿是对的，这操作起来可就难了。

## 抓住机会

让我们来破题吧！

我觉得那些心怀梦想的程序员，不安于现状的程序员，一定要考虑初创公司，这是你改变未来的一个大契机。有时你看不到未来，那就跟着能创造未来的人出发吧。

对于希望有稳定收入又羡慕梦想之光的程序员，如果一个初创公司提供的薪水还可以，那其实选择这样的初创公司和已经度过初创阶段的公司，没什么太大区别，不妨考虑一下，毕竟初创公司有很大机会快速成长，这样的机会不是天天有的。至于风险，也没什么不可承受的，顶多过两年再找工作而已。再者，你现在看着稳定的公司，也可能转瞬就裁员了、崩塌了，这年头的环境，意外太多了。

好了，终于破题了，压力骤减，一下子轻松多了。现在看看怎么选择初创公司吧。

方向这东西，你今天觉得这个正确，明天觉得那个正确，你的看法是随着生活阅历和认知提升而变化的。

看人这件事儿，那就更难了。

对于人，最好是熟人，或者是熟人推荐。程序员选择初创公司是这样，初创公司选择程序员也是这样。基于熟人推荐，有一个非常大的好处，就是可以跳过陌生人之间的信任建立过程。

对于方向，那就看你的感觉了。

考虑创业公司的方向，有几个问题可供参考，你可以问自己，也可以问创始人或公司的合伙人：

- 做什么事儿
- 行业格局怎么样
- 这个事儿能做大吗
- 切入的时间点怎么样
- 我能干什么

你还要多问自己一个问题：我喜欢这个公司做的事情吗？

如果你跃跃欲试，准备到初创公司寻找机会，那我的建议是：

不要太挑剔，方向合你心意，动心了就去干吧。机会和能力都是有保质期的，非要一切条件都确认了才做选择，那时可能已经晚了。机会这东西，总在你自己以为一切准备就绪的时候，过期不候。

你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 任性，春节前辞职

有些公司会在春节前释放岗位出来，不过说实在的，春节前招人有一些困难，你会发现大部分候选人的答复都是说要春节后再考虑。这也可以理解，辛辛苦苦干了一年了，拿到年终奖再说吧，年底加薪结果出来再说吧。大部分程序员是这种心思，不过也有一些哥们儿会很任性，春节前就辞职，这是为什么呢？

改变，从今天开始



“拖延症”是指自我调节失败，在能够预料后果有害的情况下，仍然把计划要做的事情往后推迟的一种行为。

年会还没开、年终奖还没发、年前招人的公司少……这些托辞会让相当一部分人把找工作的事儿往后拖，拖过年再说，无论之前再怎么把公司当火坑，再怎么对自己怒吼“我再也不想在这儿待下去了”，还是会夹着尾巴度过这段时间，努力装作忙碌的样子，生怕年终奖拿少了。

可有一些人不是这么想的，既然已经厌烦了此时此刻所在的公司，身在曹营心在汉，干什么都难以像以前那样投入，效率会急剧下降，哪怕你很有职业操守也很难站好最后一班岗。要走的心定了，不自觉地，该推的事儿就推了，原本会努力的事儿也都松懈了，虽然有时觉得没有以前那么敬业有点对不住公司，可还是会下意识地这么做。这是天性，不是思想、道德的问题。

工作其实和谈恋爱差不多，双方不合拍了，表面上再怎么装得和谐自然也无济于事，对于彼此都是一种负担。所以有的程序员快刀斩乱麻，不考虑现在是什么时候，即便快发年终奖，也还是拍拍屁股起身走了。

改变，要从今天开始。一旦你已经下定决心了，多待一天都是浪费青春。不但浪费自己的时间，也耽误公司的事儿。

## 有机会谈到更高的薪水

嗯，想想是不是这样子？

如果一个公司着急在年前招人，要你在年前到岗，那一定是产品或者团队遇到坎儿了，切切实实急需你来给他们补充开发力量，需要你帮助他们在预计周期内把产品开发出来。这个时候，对公司来讲，时间是第一位的，钱是第二位的，公司可以出更多的钱来吸引你。不够？我加，我再加。这么一加，每个月多出来的薪水，过上一年半载，就超过你在上一家公司那里能拿到的年终奖了。况且，薪水是月月有的。

你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

年后红海年前蓝海

## 第100000届招聘会



被称作“金三银四”的三四月份是求职旺季，拿了年终奖的兄弟姐妹们心思活泛了。其实这不是季节性的，而是已经形成的一个现象了。

这种现象，抛去年终奖的因素不谈，其实也是有现实依据的：新年一开始，公司在人资上有预算，哪个部门要招几个人，都是在年底年初做预算的时候定下来的。这也是到了三四月份，各公司的 HR 们满脸春色向“阿猿们”抛媚眼的一个原因。

大家都知道的秘密，那就不再是秘密了。大家都知道三四月份机会多，那反过来，其实找工作的人也多，好的公司、好的岗位，应聘者会比平时多好多倍，用门庭若市来形容毫不

为过。HR 这时就会多面几个人来比较一下，看看有没有更合适的人选。这也是招聘一方的心思，市面上鱼多了，当然要挑一条卖相好的了。

对于程序员来讲，这其实是求职的“红海”，诱人的岗位竞争者众多，反倒不容易脱颖而出。而年前因为大家都在等年终奖，找工作的人反倒少了，再加上公司预见到年后很多人会孔雀东南飞，还会预先释放一些岗位出来，

再说一点不出奇的，大家别笑话。那些能力很强的程序员，年终奖往往是丰厚的，他们很多人都舍不得放弃这个，有换工作的想法，也是要等年终奖到手再说的。对于我们这些二流角色，原本只能做备胎的，如果敢于放弃年终奖，那其实是更有机会的。因为年前实际上是招聘市场供需失衡的时候，一流的找不到，那就二流的，二流的还没有，那就三流的……有些公司这么一琢磨：找三流的人，给二流的钱，干一流的活，太划算了。所以他们也乐于给你比三四月份更多的薪水。

## 年后变数多

还有一点，其实年后会比年前有更多的变数。比如你原来决定了要走，可是一看年终奖还不错，那是不是我再干一年？比如我阿猿是单身汉，年后忽然公司来了几个做 UI、做测试的女同事，那我是不是机会来了？比如老板为了留人，把年终奖分多次发放。又比如老板为了留人，答应过段时间给你加薪升职……

诸如此类，如果你没有关二爷的定力，面对美女、名车、金元宝、升职，往往是会忘了要飞的想法或者延迟了高飞的时间。

其实，我们换工作，更多的考虑，应该是顺应自己职业发展的需要，是着眼于职业生涯的规划，是想着一年更比一年好，是自然而然的时间到了，而不是“等年后了我来考虑这个事儿”。所以，有些“阿猿”这么想的：在哪个时间点我的天使降临了，那我就那个时间点跳。其他因素带来的都是一时，好机会带给我们的是一世。

啰啰嗦嗦这么多，其实我想说的是：

晚上想想万条路，早上起来走老路，最后的结果是没路。所以，谋定了就出发吧，只有前方是值得期待的。

## 让程序员跳槽的非钱原因

程序员工作仅仅是为了钱吗？

NO!

工作的一个很重要的目的是获取薪水，获取更多的薪水。但，这仅仅是一方面，很多时候，决定你是否愿意在一家公司持续工作下去的原因和钱无关。

马云说员工离职的原因林林总总，但基本上可以归结为两点：1. 钱，没给到位；2. 心，受委屈了。这些归根结底就一条：干得不爽。这一节咱就来看看程序员会因为哪些不愉快而跳槽，与钱无关的那些。



## 让人无语的领导

相当一部分人离职，是因为他们觉得领导很不专业，也不称职，不配领导自己，在这样的领导下，工作让人绝望。

有些领导通常不把程序员当回事儿，认为程序员就像机器，只要按吩咐干活儿就行。他们通常不尊重程序员和程序员的劳动。比如他们可能会说这些让人难堪、窝火、愤怒的话，丝毫不考虑程序员的面子。

“又请假，活儿干完了吗？”

“我觉得这个需求简单得很，一天搞定。”

“干不完就加班，通宵也得搞定，明天一定要上线！”

“干活儿不动脑子啊！”

“活儿都没干完，还一下班就走，一点儿责任心都没有！”

### “没见过你这么笨的家伙！”

有的领导喜欢给程序员扣帽子。

比如网站新版上线后，购物车功能不正常，要添加的商品超过 4 个就崩溃，上级领导问起，某些领导就会说：“购物车是张三负责的，估计什么地方没注意，我马上找他确认。”

比如上级领导问项目为什么延期，某些领导就会说：“张三李四效率太低还不愿意加班，卡在他们那儿了。”

还有的领导，就是个传话筒，上级领导说网站要改版一个月后上线，他一口就应承下来，答应一个月后上线。然后转过身来就吩咐这帮苦逼的程序员：“领导要求一个月内必须搞定，大家辛苦点儿，加班加点也要搞定。”

你有没有评估过可行性啊，最起码应该问问这帮程序员啊！



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

还有的领导天天都表现得像更年期的祥林嫂一样，一天能跑过来问你十次：“阿猿，怎么样，什么时候搞定？”

还有的领导老怀疑你不干活儿，经常假装走过你身边，瞄瞄你的显示器，然后问：“你干嘛呢？”

还有的领导自己很轻松，天天早早下班不说，临走了还给你来一句：“我先走了啊，各位辛苦。”

还有的领导自己的事儿也推给别人干，“张三，帮我写个月报。”，“李四，帮我汇总一下项目状态。”，“小三，我明天去客户那里交流，你帮我做个介绍项目的 PPT。”，“阿媛，告诉大家今天加班。”，“小刘，告诉王五让他晚上盯着服务器，美国那边要演示，千万不能出问题。”，“小四，打电话问问小三为什么到现在还没来。”……

还有的领导只关注你的错误，你犯错了他们就会揪着不放。你干得好，他们就认为是理所应当的，上级领导表扬时，丝毫不提程序员们的贡献。比如你提了一个可以避免 90% 进度问题的过程改善方案，他向领导汇报时居然一句也没提到你。比如公司都知道项目被卡在 Windows 7 的 64 位驱动蓝屏上了，老板也知道了，你解决了这个问题，老板表扬你们团队，但领导并没有提你的名字。

还有的领导，总是让你自己去解决应该由他解决的问题。比如你告诉他“电脑太慢，编译一次 Android 系统要 10 个小时”，他就让你去想办法，诸如晚上编译、添加-j4 参数什么的。比如你要开发 iOS 应用，老板不舍得配 Mac 电脑，他就让你去装黑苹果，还限期搞定。比如你告诉他需求三天两头变希望他想办法改善这种状况，他就会说这都是不可避免的，开发应该拥抱变化……

还有的领导，分不出谁能干谁不能干，甚至有时会很亲近那些偷奸耍滑的家伙。

一个不合格的领导，会把一个团队带向深渊，有追求的程序员会因之一个一个走掉。

## 死水文化

如果一个团队缺乏朝气，死气沉沉，大部分人得过且过，当一天和尚撞一天钟，那也很

难留住有追求的程序员。

比如你看到公司的认证服务器采用的方案是“一个线程处理一个客户连接”，这种方案很难应对高峰期的大并发访问，找架构师讨论是否该用 NIO 或者 MINA 来重构一下，人家都不愿意搭理你……

比如你觉得公司的商品搜索服务每次都返回所有结果，而手机上一页只能显示 10 个，当结果超过 200 个时解析就会很慢，超过 1000 个时简直就不能忍受，你去找实现搜索服务的哥们儿，他告诉你说“需求没提，别管它”，你去找开发手机客户端的哥们儿，他告诉你“先这样吧，要实现分页查询又得改接口又得改逻辑，太麻烦了”……

比如你在用公司的 Android 客户端时发现商品展示页面，翻页多了程序会崩溃，猜测可能是图片占用内存太多出现了 OOM 错误，可是负责客户端的哥们儿说，“用户绝对不会像你那样一直翻页的”……

当大家对代码和产品没有追求没有感情觉得怎么着都成时，追求完美的你，就会感到格格不入，这样半死不活的团队，你是不愿意待下去的，除了离职，还有别的选择吗？

## 猪队友

有些领导没办法识别团队中坏了一锅汤的那粒“老鼠屎”，这会让兢兢业业勤勉上进的其他成员感到不公。

不怕神一样的对手，就怕猪一样的队友。

## 绝望的产品

当你遇到这样的产品，方向不明，市场不明，号称要跟随市场脚步，找不到自己的定位和核心竞争力，各种乱改需求，程序员们像堂吉诃德一样可笑地战斗，你的心情是怎样的？

路越走越漫长，看不到一丝光亮，心越来越绝望。

你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 发展空间



团队里没有高手，连个切磋的人都没有，这是何等寂寞的事。遇到问题没人能搞定，又是何等无奈。

你只要完成上头交待的任务就行了，你的技术之路如何发展，你在公司的职位和等级怎么晋升，这些从来没有人找你谈过，也没有人真的关心你。

你只要干好你手头的那些工作，其他的你不要管。

你只要熟悉这项技术，每次分配任务都根据你现在掌握的技术来定，你只能干你能搞定的活儿，不给你接触新技术的机会。当你决定要用一个新技术来实现某个功能时，领导就会告诉你完成任务是最重要的，冒险是不允许的。

要知道，很多程序员之所以愿意做开发，就是有强烈的学习兴趣，愿意不断挑战自己。

失去学习和锻炼的机会，在这里待着还有什么意义呢？

要知道，每个程序员都希望自己的竞争力越来越强，眼看着只能拿个锤子敲一辈子钉子，谁能压抑住野草一样生长的忧伤与绝望？

## 薪水倒挂

有的领导很奇葩，宁愿花更多的钱请新人，也不愿给老员工涨工资，以至于同样的工作经验，新来的人居然工资比老员工高一大截，这叫辛辛苦苦兢兢业业的程序员情何以堪。

这不是钱的问题，而是领导的观念问题。

## 没有来由的加班

假如要配合某个市场节点，加班加点赶进度，这可以理解。

假如线上产品有重大 **Bug** 曝出，加班加点 **Hot Fix**，这可以理解。

假如我是新人，还不能胜任目前岗位，加班加点能提升自己，这可以理解。

假如我负责的模块因为我的个人原因进度落后，可能导致项目延期，加班加点向团队看齐，这可以理解。

假如竞争对手突然曝出一个彪悍的功能，我们的用户大量流失，产品必须迅速跟上，加班加点挽回损失，这可以理解。

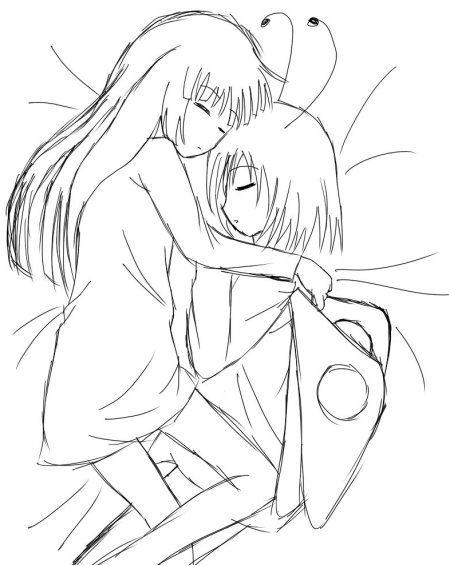
假如只是老板觉得看不见人加班心里不爽，这让人怎么接受？

假如领导总是错误估计工作量，导致每次都要加班加点为领导的轻率买单，这让人怎么接受？

假如领导安排任务时就把加班加点算计在内，这让人怎么接受？

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 无 Bug 不生活



我决定谈一谈世界上最著名的虫子：Bug！

Bug 困扰了一代又代的程序员，不论是杰出的计算机科学家，还是像 Linus Torvalds（Linux 内核创始人）、Bill Joy（传说三天写出 BSD 的前身，vi 和 csh 的作者）等神一样的传说，抑或你我芸芸“程序猿”，都是 Bug 骚扰的对象。Bug 是绝对狂热的好战分子，具有永不停歇的战斗欲望，它潜伏在程序员的周围，一双小眼贼亮贼亮，在你百密一疏时出其不意、一击奏效。而无论你是钢筋铁骨，还是羊脂玉体，只要被这只虫子袭击，就得蜕掉三层皮，更有甚者，很可能锒铛入狱、命丧黄泉或者烟消玉殒。

### 软件界致命的 Bug 事件

其实，Bug 造成的悲剧天天都在上演，这里我们先举几个比较著名的例子。

## 许霆案

2006 年 4 月 21 日晚 10 时，许霆来到天河区黄埔大道某银行的 ATM 取款机取款。结果取出 1000 元后，他惊讶地发现银行卡账户里只被扣了 1 元，狂喜之下，许霆连续取款 5.4 万元。当晚，许霆回到住处，将此事告诉了同伴郭安山。两人随即再次前往提款，之后反复操作多次。后经警方查实，许霆先后取款 171 笔，合计 17.5 万元；郭安山则取款 1.8 万元。事后，二人各携赃款潜逃。

从软件的角度来讲，其实是 ATM 提款机出现了一个 Bug，许霆作为测试人员发现了这个 Bug，然后找来朋友做了多次复现操作。

这是一个软件 Bug 引发的悲剧，最终上升到了犯罪的高度。

2007 年年底，经过一审，广州市中级人民法院判处许霆“无期徒刑，剥夺政治权利终身，并处没收全部个人财产”。

2008 年 1 月，广东省高院裁定许霆案发回重审。经过再审开庭，法院做出了有期徒刑五年的判决。

作为程序员，我认为开发 ATM 机的公司应当负有责任，他们应当赔偿客户（即银行）的损失，是他们的软件 Bug 导致了这个案件。但在银行这种强力机构面前，个人的声音微不足道，很难把两者放在同一个天平上来考量。这中间到底有着怎样的曲折，时过境迁，我们已无法考量。但是，作为程序猿，我们还是得引起警惕啊，很多时候，Bug 会带来直接的经济损失和严重的法律后果。虽然你可以辩称技术无罪，但子不杀伯仁，伯仁因你而死。

## 锤子手机预约数闹剧

2014 年国庆假期期间，锤子手机 4G 版现身天猫商城，页面显示该机预计将在 10 月 18 日正式开售，售价为 3500 元。据说备货量为 10000 台，很快天猫商城页面显示预约人数超过 8 万。

后来，有网友爆料称从页面源代码中发现 4G 版锤子手机的预约人数有水分，页面显示的数字是实际预约人数的三倍，从而再度把锤子手机推到了风口浪尖上。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

后来，天猫对这一事件做出了正面回应。天猫表示，截至 2014 年 10 月 7 日，锤子手机 4G 版的预约用户数为 62682 人，但在 2014 年 10 月 8 日，系统调用一个数据端口时，意外将前端该页面动态数据显示“清零”。

为了尽可能让预约数据还原真实，天猫决定将 2014 年 10 月 8 日起的新预约数做“乘以三”的处理，以便后期预约数能快速接近真实数据……

注意我设置为斜体的那句话，如果天猫的回应符合实际情况，那么，其实，这是一个 Bug！

2014 年 10 月 19 日，老罗在微博上正式回应：预定数据造假的事，跟我们没有一丝一毫的关系，我们在这件事中扮演的唯一角色，就是“躺枪”。

这是要上演罗生门的节奏吗？

对于此事，我不能不感慨：不怕神一样的对手，就怕猪一样的队友；不怕猪一样的队友，就怕神一样的网友——据说这个造假的 Bug 是因为某个测试人员发现预约数总是 3 的倍数而发现的。

## 苹果 iOS 8.0.1 更新撤销

苹果，神坛上的苹果，以极致用户体验而载誉江湖的苹果，也因为 Bug 而被钉在了耻辱柱上：

“(2014 年)9 月 25 日消息：苹果在今天早些时候发布了 iOS 8.0.1 更新，目的是解决当前版本中的多处 Bug，但是没想到的是更新补丁带来了更多的 Bug，最终苹果撤回了该次升级。

“据了解，本次受到影响的主要是 iPhone 6、iPhone 6 Plus 这两款最新的设备，更新后无法使用蜂窝网络及 Touch ID Home 键。因此建议还没有升级的用户还是不要升级了，而已经悲剧了的用户目前只能等待苹果推出修复补丁。”

有人说这是乔布斯不在的后遗症，真不知道这帮人为什么总是把乔布斯拿来说事儿。其实，这就是一次软件 Bug，从“程序猿”的角度来看，这再正常不过了，和乔布斯有什么关系呢？

再说了，这事儿又不是单单苹果一家才有，你看人家微软，根本都不当回事儿，补丁一

一个又一个又一个又一个地发，堵的就是漏洞，改的就是 Bug，这都是那帮“程序猿”惹的祸，与天上的月亮没有关系。

## 心脏滴血漏洞

“2014 年 4 月 9 日，Heartbleed（意为“心脏出血”）的重大安全漏洞被曝光，一位安全行业人士在知乎上透露，他在某著名电商网站上用这个漏洞尝试读取数据，在读取 200 次后，获得了 40 多个用户名、7 个密码，用这些密码，他成功地登录了该网站。

“Heartbleed 漏洞，这项严重缺陷(CVE-2014-0160)的产生是由于未能在 memcpy() 调用受害用户输入内容作为长度参数之前正确进行边界检查。攻击者可以追踪 OpenSSL 所分配的 64KB 缓存、将超出必要范围的字节信息复制到缓存当中再返回缓存内容，这样一来受害者的内存内容就会以每次 64KB 的速度进行泄露。”

OpenSSL，著名吧，开源吧，安全吧，它支撑的互联网站点就像天上的星星一样，数都数不过来。就这么一个知名、开源、强大的密码库，一样有 Bug。

Bug 无处不在！

Bug 对“程序猿”的追逐，对软件的痴迷，比现在的死忠粉还要厉害，那简直是上穷碧落下黄泉，不论你在哪里，上天入地，我都会与你在一起，不离亦不弃。

## Bug 的定义

Bug 一词的英文原意是“臭虫”或“虫子”。但对程序员来讲，Bug 意味着错误、缺陷、问题、漏洞、未按需求实现、不符合需求预期或延伸等等……那么，Bug 究竟是什么呢？

我爱人是搞测试的，软件测试这个领域最著名的一本书就叫作《Software Testing（软件测试）》，它里面是这么描述 Bug 的：

“The software doesn't do something that the product specification says it should do.

The software does something that the product specification says it shouldn't do.



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

The software does something that the product specification doesn't mention.

The software doesn't do something that the product specification doesn't mention but should.

The software is difficult to understand, hard to use, slow, or in the software tester's eyes will be viewed by the end user as just plain not right.”

作为开发人员，当然要看得懂英文了。

其实还有一个粗暴点儿的说法：Bug 就是错误。

## Bug 是怎么来的

《圣经》中有“义人”这个说法，我们有“人无完人，金无足赤”的说法……

人是妈妈生出来的（女娲造人时除外）——你看我总是加各种说明以便堵住漏洞，这样写文章真是很累。这实在是和我的开发习惯有关：我总是习惯防御性编程，预先考虑各种错误，在编码期间就努力消除它们发生的条件。

我们说，人是人他妈生的，妖是妖他妈生的，而软件是程序员生的。

软件是程序员的孩子，在生它之前，你可曾封山育林、可曾熟悉自己的生理周期？开发软件其实也要经历这些过程，更重要的是，我们一定要意识到程序员与 Bug 密不可分的关系。

## 程序员在生产软件，也在生产 Bug

再牛的程序员，写出来的软件也有 Bug。Bug 就是软件与生俱来、不可脱逃的“好朋友”，就在那场说走就走的旅行的一开始，它们就情根深种难分难舍了。

所以，Bug 是程序员写出来的。

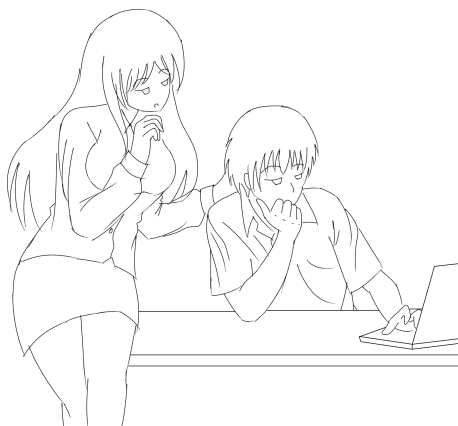
不管你愿不愿意承认，事实就是这么令人沮丧。连 Maurice Wikes 都承认了这一点，你就别倔犟了。

## 程序员对 Bug 的态度

每个程序员都是不同的。

绝对不可能……

点这里，回车，就崩溃……  
咦……



测试 MM 对阿猿说发现了一个 Bug。

阿猿矢口否认：不可能，绝对不可能！

MM：真的有 Bug，你过来看一下！

阿猿：我都不用看，在我这儿好好儿的。

MM：你来看一下嘛。

阿猿：看什么看，我都测过的，绝对没问题。

伴随着凳子响，阿猿不情愿地起身，脸上不耐烦的表情，心里却嘀咕着这是怎么回事儿。

☞ 你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

过了一阵儿，阿猿不好意思的声音响起：真没想到是这儿的问题……你怎么测的……一般用户都不会这么用的……

耶，你找到 Bug 啦

测试 MM 袅袅婷婷地走到阿猿跟前，小声说：你来我这里看个现象。

阿猿：好，马上来。

阿猿立马起身跟 MM 走了，过了一会儿，响起阿猿的声音：哇，你太厉害了，这种问题都被你发现了，我测了很长时间都没测出来，太好啦！

然后大家看见阿猿脚步轻快地走向工位，边走边频频点头，两只手像得了帕金森一样来回划拉，嘴里神神叨叨地念念有词……

结果到工位上时，阿猿一拍额头，小声说：知道啦，原来是这样。

过了两分钟，阿猿找 MM 说：问题解决了，是这么这么回事儿……

MM 一脸崇拜地看着阿猿：你太厉害了，这么快就解决了！然后小声说：大家都像你好啦……

阿猿说：有什么问题立即叫我看。

我很忙，没空管 Bug

测试 MM 想在 jira 上提个 Bug，先在 QQ 上对阿猿说：有个 Bug，你过来看一下？

阿猿：忙着呢，焦头烂额的。

MM：一分钟都用不了，你来看下吧。

阿猿：思路一打断就难恢复了，等会儿！

MM：你不看我提到 jira 上了啊。

阿猿：随便，你不就是爱提 Bug 嘛。

.....

一天，团队 review Bug，TL 问阿猿：这个 Bug 怎么回事儿？

阿猿挠挠头：我也不知道啊。转身问 MM：咋回事儿，提 Bug 也不通知我一声儿，我都不知道怎么回事儿。

MM：当时给你说你没时间看，后来……

阿猿：行啦行啦，我待会儿就看，以后提 Bug 要通知我一声。

MM 无话可说，只能用满含幽怨的眼睛锁定阿猿，直到阿猿别过头去……

## 都是需求的错吗

阿猿收到了来自 jira 的邮件，他被告知有一个 Major 级的 Bug，看了之后，黎叔很生气，后果很严重——阿猿转身就喊：MM，怎么回事啊你，给我提了个 Major 级的 Bug，明明是需求问题啊！

MM 赶紧过来确认：哪个 Bug？

阿猿气呼呼地一指电脑屏幕：就这个，需求问题，你仔细看看，别老给我整这种 Bug。

MM：咱们找需求文档看看……

MM：你看，需求是这样描述的……

阿猿：需求描述有问题，实现起来太复杂了，也没意义，谁会这么用啊。

MM：我是按需求写的用例，要不咱们和产品经理一起讨论一下？

阿猿：算啦算啦，麻烦，先不管啦，到时再说！

## 怎么会还没好

begin:

MM：阿猿，你提交的版本有问题，Bug 还没好。

☞ 你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

阿猿满脸自信：怎么会，我都测过了，各种情况都考虑到了。

MM：真的，不信你来看。

阿猿：看就看，你告诉我你怎么测的。

MM：这样进来，这点一下，马上再点一下，趁那个按钮还没刷出来，再点，然后就报“XXX 已停止运行”。

阿猿：用户不可能像你这么用！

MM：可能。

阿猿：不可能，我从来没这么用过。

MM：我先给你 Reopen，你再改改吧。

阿猿：不要 Reopen，根本没问题。

MM：我都遇到了。

阿猿：算啦，你别 Reopen，我先改改，马上给你一个新的版本。

过了几分钟，阿猿提交了版本。

goto begin;

## 测试环境没问题

经过多次延期，经过几个晚上的集体猛搞，软件终于发布了，大家都送了口气儿，第二天都休假去了，只有阿猿留了下来。

刚上班，阿猿就被气急败坏的老板叫到办公室：赶紧把版本回退回去，摊上大 Bug 啦！

阿猿一脸无辜：测试都没问题啊，什么 Bug？

老板：两个小时，升级了 800 多个用户，有 40 多个电脑死机的！

阿猿：不可能啊，测试环境都测过了……

老板：赶紧停止升级，下线版本！

老板还说：你们就是太大意了，没有好好测，都不当回事儿。

阿猿：其实大家也是蛮拼的，都搞了好几个通宵啦……

老板：赶紧下线，拼不拼不重要，没问题是目标。

阿猿：要不我们联系几个用户，升级测试一下，重现了就好定位问题！

老板：哪个客户愿意当小白鼠？赶紧下线！你们自己想办法，多用点心，别不当回事儿，我们都被告到消协啦，这样下去早晚出问题。

阿猿：测试环境真测不出来……

阿猿面对老板喷火的双眼，心有不甘却只能无奈地转身出了办公室，嘴里念叨着：什么毛生产环境总是出问题呢？

## 怎样才能告别 Bug



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

怎样才能告别 Bug?

有人说提高代码质量，参见《代码大全》。

有人说提高自我修养，参见《程序员修炼之道：从小工到专家》。

有人说提高效率，参见《高效程序员的 45 个习惯》。

有人说……

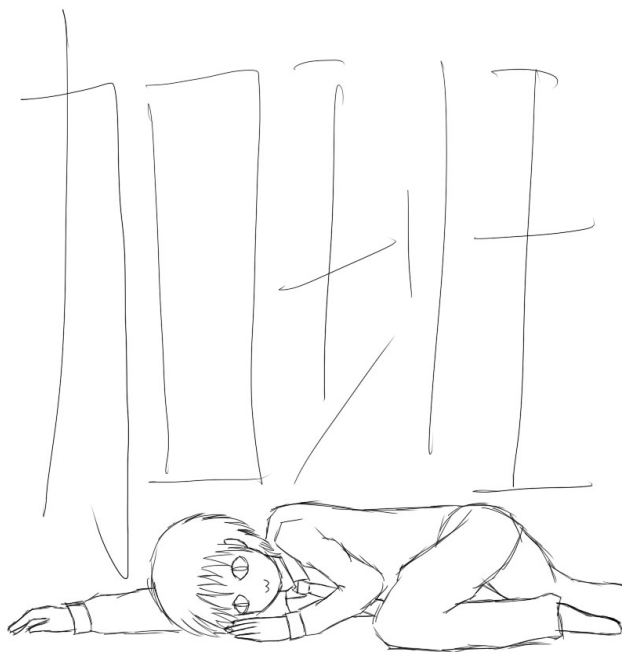
优秀的软件也会随着外部环境的变换而老化，最终不能满足用户需求或者无法按预期工作。

计算机先驱 **Maurice Wikes** 回忆起 1949 年他在英国剑桥工作的情形，在拖着打孔纸带上楼给雏形计算机 EDASC 装载程序时，他看到了自己的未来：

我强烈地意识到，生命中剩下的好日子，都将耗费在给自己的程序找错误上面。

程序员正在用自己的生活，给 **Maurice Wikes** 写注脚。

## 谁是为加班而生的



有一个问题经常在面试的时候被问到：对于加班，你怎么看？

大家都知道 IT 行业加班很多，不加班的公司风毛麟角。很多公司都是加班出了名的，比如 X 为、X 米。还有很多公司的老板认为自己的公司之所以业绩不行，就是自己的员工不能像别的公司那样加班。

你加班吗？有加班费吗？这是我们程序员之间的闲聊话题。

你面试时，面试官通常会问你怎么看待加班，我们怎么回答呢？



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 我的父亲说

我的父亲是这么看待加班的：反正在家也没什么事儿，再说加班说明公司有业务，有业务说明公司有前途。

这是很多人的看法。

## Work-Life Balance

上中学时经常听到一个词叫作“体验生活”，比如某某领导到农村体验生活，某某作家到监狱体验生活，比如学生应该多多体验生活……好像我们当下不是在生活，好像只有更苦更不堪的日子才叫做生活。

中学时写作文，老师有时会这么批：缺乏生活，没有真情实感。

我们读小说，有时这么议论：矫揉造作，没有生活基础。

那到底什么是生活？

其实我们天天都在生活，不需要为生活单独定义，只要你活着，就在生活。生活方式有千千万万种，比 32 位无符号整数能够表达的数字还要多。

那工作是不是生活呢？

我们每天有 24 个小时，至少 8 个小时用于工作。假设我们可以活 75 岁，65 岁退休，25 岁工作，工作年限有 40 年。那么工作的日子就有 13 年，甚至更多。假如每天睡 8 小时，那么睡觉时间为 25 年。3 岁上学，25 岁毕业，上学有 22 年，减去 22 年中的三分之一睡觉时间，还有 15 年。好啦，就这么简单吧： $75 - 25 - 15 - 13 = 22$ 。就是说，我们有 22 年在生活，再刨除小时候的三年，太小了，不懂生活，剩 19 年。老年的时候会被各种病痛折磨，将死未死那几年也没什么生活质量，儿女觉得你比狗都讨厌，再刨除 5 年吧，还剩 14 年。

能这么算吗？

.....

……此处省略 500 字，请思考一下……

……请在这里停下来，想想你的生活……

其实工作就是生活的一大主题。认为工作是生活对立面，多数时候是因为工作没有乐趣，或者工作之外的很多事情没时间处理。

## 加班的分类

从某个角度来看，有三种形式的加班：

1. 充满希望的加班；2. 绝望的加班；3. 习惯性加班。

### 充满希望的加班

有时候一个产品，多花一些时间，早一天推向市场，快人一步赚钱无数，此时公司会要求大家加班赶进度。你可能会反感，但是想着明天就可以数钱数到手抽筋，心里也不会特别抵触。毕竟，工作的一个主要目的是获取薪水。你还会想着，等产品上线就好了，就没那么忙了，领导也是这么说的。总之你认为这是临时性质的加班，不是制度性的，忙过这阵子就好了。这种加班在多数人看来不可怕，有希望，就有动力，人为着一个希望去努力去拼搏，通常会给自己灌迷魂汤。

有时我们很相信这种希望，不是有首歌叫《爱拼才会赢》吗？我在每个公司做产品的时候，都跟我爱人说，这个产品很有前途，加班只是暂时的，产品上市就好啦，你看，我已经患上斯德哥尔摩综合症。

### 绝望地加班

产品或者项目已经明知道要失败，还是要给客户一个交代，或者让客户知道，我们已经尽力，是力战而死，而非有意退却仓皇逃窜。此时我们也会加班，大家都心知肚明，就算是再努力，再多拼上一段时间，对最终的结果也没什么改变。本来嘛，注定要失败的。可是不得不，还是要加班，绝望地加班。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 习惯性加班

当爱已成往事，你却还保留着昔日恋人陪你养成的那些习惯。既已成习，改之谈何容易。我的一切，都发生在这里，我在这里开始软件开发的生涯，我在这里开始半推半就的加班，我在这里终于把加班培养成了习惯。久而久之，不加班竟有点不习惯，不知道可以干点什么，心里没着没落，安定不下来。

## 加班的可能原因

没有无缘无故的加班。

## 因为喜欢，所以付出

当你爱上一个人，不会去比较她爱你是否比你爱她少一点。因为爱是给予，是付出，是想对方好，而不是对等的交换，不是授人一滴水望伊涌泉报。

有的人很幸运，会喜欢上他的工作，对他来讲，工作是一种享受，是一种快乐。那这事儿就好办了，带给人享受和满足的事儿，自然愿意多干一些，再多干一些……

## 博取好感

娱乐界比较混乱，很多人靠事业线搏上位。IT界比较简单，有些程序员想搏上位会自然而然想到加班。有时你看“阿猿”嘿咻嘿咻搞一晚上都不肯停下歇一歇，其实是因为刚好今天领导在……

## 企业文化

大家都这样，你要不这样还真不好意思不是？

## 工作效率低

也有的同志，天生难以集中注意力，也难以按正确的方法做事，因此干起来事倍功半，

谁是为加班而生的🦉

大家伙觉得 5 个小时就能搞定的事情，他要搞 12 个小时。我们评估工作量是取均值，领导又只看工期和结果，所以只好加班。

## 项目需求变化

程序员最恨的不是近在眼前的 Bug 寻寻觅觅寻不到，而是干到一半产品经理忽然说需求变了。

## 消磨时间

还有的朋友下了班没事儿可干，回到家里也是玩电脑打游戏看视频，不如在单位混个加班，还能有个餐补、调休什么的，真是太划算了。

## 家里有怪兽

有个哥们儿，天天加班，不觉疲累。

好事的同事就问他：“阿猿，你天天加班为那般呢？”

阿猿不好意思地说：“嗨，娃一岁多点儿，天天晚上看什么花园宝宝，有点儿受不住……”

原来如此！等阿猿加完班娃也看完花园宝宝睡觉了，刚刚好。

## 怎么看待加班

我们的问题还是没能绕过去。

其实，在正常的工作时间内高效地完成工作是最值得推崇的。

我们所期望的生活是：做自己喜欢的工作，事业有成，同时有自己的时间去休息、锻炼、学习，和家人在一起。

那么，去学习如何高效地工作吧。

💬 你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 别说我已经很努力了

我们程序员的努力与挣扎有时非常尴尬，如果没有结果，都是徒然。



我自己做项目经理时，干的项目也经常延期，非常惭愧。而延期其实对研发团队、产品线、市场、销售、公司领导、客户、用户都是一种折磨。对其他人的折磨这里就不说了，谁被虐谁知道，至于有没有想死或者想消灭谁的心就甭去管它了。作为程序员，我其实也只能体会到延期对程序员们的折磨。

### 也是蛮拼的

在相当长的一段时间内，“也是蛮拼的”这个短语在网络上大行其道，我们先看看它的出处。

先做“剪报”吧，这是我的强项。来自百度百科：

别说我已经很努力了🐼

“也是蛮拼的”，出自爱奇艺热播剧《白衣校花与大长腿》，然后通过《爸爸去哪儿2》曹格之口使其发扬光大，在网络上被大家所熟知，并广泛传播。

“也是蛮拼的”意思是：挺努力的意思，但是即使很努力了，却没有成功，具有反讽刺意味。

本文就取这个意思。

## 当项目延期时

我带的项目，很少有按期完成的。

当项目延期时，领导一方面很怒，心中不满，嘴上却要尽量安抚。领导有时面对项目的延期，是干着急没办法，既愤怒又无力。

此时项目经理也很无奈，大家都拼了几个月了，天天加班到十一二点，很多人把夫妻恩爱缠绵都免掉了，到临上线还要通宵，结果还是不行，上线了又出问题然后再被发回重审回炉再造。领导两眼血红只盼三几分钟就能完成 Hot Fix，可是项目经理说：其实大家也是蛮拼的。

领导听见这话烦躁得很。

项目经理又说，大家已经很努力了。

领导再也按捺不住了：“努力有什么用，你的工资是论努力给的吗？不解决问题，任何的努力都是借口。甭拿努力当挡箭牌！你们就算天天打游戏，只要出活儿比什么都强！”

项目经理：“没人打游戏。”

领导：“我是打个比方！别纠缠细节！”

项目经理：“其实大家真的蛮拼的，天天晚上加班，经常工作到凌晨。”

领导：“你们不努力行吗？难道我发给你们薪水，是让你们端着茶杯看报纸的？再说，

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

努力是工作的基本要求！”

项目经理：“对，工作本来就应该努力，其实大家都知道。”

领导：“都知道为什么活没干好？是不是这帮人能力不行？你划拉一遍，谁不行就开掉，招一个新的进来。当然你的能力是大家都清楚的，我是说总有几个人上班时打游戏、聊天、刷微博、上淘宝，都没事儿干，你知道吗？”

项目经理：“其实也不能把大家逼得太死，那样……”

领导：“我逼你了吗？”

项目经理：“我是说，从项目管理的角度来讲，应该提出要求，关注结果，让团队成员自己寻找达到目标的路径，没必要规定路该怎么走，只要完成就行。每个人都有自己的工作方式，让他们发挥自己的长处。”

领导：“项目完成了？”

项目经理：“一开始给的工期就有一些问题。”

领导：“为什么不早说？”

项目经理：“……”

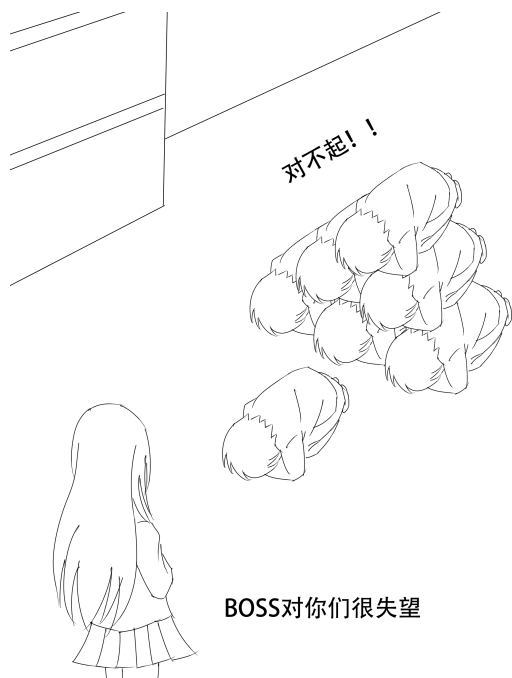
领导：“再给三天时间，一定要保质保量地上线！”

项目经理：“好。”

努力为什么说不得

“努力”原本是个褒义词。

小时候老师给的期末评语，如果写上“此



子努力向上”之类的，父母都会欣慰一笑。

古训也说：少壮不努力，老大徒伤悲。

努力学习，天天向上。好多学校的墙上都刷这个。

囊萤映雪，凿壁偷光……这些词都教育学生们要努力。

虽然学生常常被教育要努力学习，要吃得苦中苦方为人上人，可因为人的才干有区别，加上天朝的教育理念落后，不贯彻因材施教、因人而异的策略，大部分学生被削足适履，再怎么努力成绩也是一塌糊涂。但为人父母的，总有爱子之心，见孩子已经“头悬梁锥刺股”了，已经努力了，也就不忍再责怪孩子。于是久而久之，孩子们都学会了：妈咪、爸比，我已经很努力了。这一招显然在父母那里比较管用，因为他们看到你晚上埋头孜孜不倦的读书（其实可能在埋头看漫画或者玩手机）到十二点，觉得既然孩子都这么拼了，结果怎么样就别再强求了。

在职场中，也有这么一种心理在作祟。

其实很多人是拿“努力”做挡箭牌：我都这么努力了，你还好意思怪我吗？

但对掏钱让你干活的老板来说：交活才是硬道理。

老板奉行的是“不管黑猫白猫逮住老鼠就是好猫”。他们可没有父母那么好脾气那么心疼你。所以，要展示结果，光说努力不出结果，难免让人怀疑你的诚意，终究会给人留下“打酱油”的印象。

当老板已经面临非常严重的交付压力，被客户把一个头骂成九个大的时候，你还在那里磨磨叽叽说“我已经很努力了”，老板估计活剥了你的心都有。

我们要谨记：没有结果时，不要随便说“我已经很努力了”这种会招致巨大副作用的废话。

## 努力要有价值

前面已经说过，只有成功的结果摆在那里时，努力才是一枚勋章。成功者总是要保持优



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

雅，哪怕一晚没睡，也要做个头发抹点眼霜让自己看起来精神一些，只要让别人看到成功，不要让别人看到背后的辛酸。要顺着这个思路干下去。这才是值得推荐的。一旦你成功了，自会有人发掘成功背后的努力，就像苏秦那样。这个时候，也就是当别人说你努力的时候，你才是真的努力了，你的努力才有价值。如果你失败了，却像祥林嫂一样叨叨“我已经努力了”，只会让听者厌烦，甚至怀疑你的能力。

多干事儿，少说“我已经很努力了”这种废话，因为越是把努力挂在嘴边越难成就，难免加强“失败也无所谓”这种暗示，暗示得多了就成了事实了。

如果真的努力了又无法完成



真有这种情况，预期可以完成，但实际无法完成。这个时候怎么办？也不能说吗？

不是不能说，是要用事实说话，努力要可见。

比如说领导让你实现 P2P 打洞，给你一星期时间。你答应下来，可是到周五的时候你没能完成。这个时候怎么办？怎么样才能让领导相信你真的已经努力了？仅仅一句“我已经很努力了”是不够的，这会让人有“尽人事听天命”的感觉，这种感觉是 **Out of Control**，是摆明了用努力为自己脱罪，领导都不喜欢。

你想想看，假如你家装修，工头告诉你水电改造两个星期完成，结果临了告诉你没弄完，你是什么感觉？这时候你肯定怀疑工头根本没尽力，一天 8 个小时可能只有 8 分钟在干活，只有他给你一个说法，比如需要 4 种水管，每种多少，需要 3 种规格的电线、8 种不同用途的开关，而这些材料应该什么时候就绪你却没提供，或者因为有特殊要求，某些材料他跑了 8 个建材市场，花了一星期时间都没买到。只有这样提供具体的原因和数据，你才能在理论上接受，虽然情感上还是会感到愤怒。

软件开发也是这样。你必须让你的努力可见，这样领导才可能从逻辑上接受任务没有完成这个事实。比如你可以说：

从类型上看，NAT 有 4 种类型，对称型、全锥形、限制锥形、端口限制锥形。我周一完成了类型分析，设计了对称型和全锥形的穿透算法。

周二完成了限制锥形、端口限制追星的算法。整理了 4 种类型的穿透算法，写了文档。发现需要一个配置公网地址的服务器来辅助完成穿透。

周三我找网管协调服务器，网管说没有设备，需要申请。我绘制了 Udp 打洞和 Tcp 打洞的流程，完成了服务器端的代码。

周四....周五....

假如你像上面那样说，让你的努力可视化，那你就可能得到谅解，也可能得到进一步的支持。因为你所做的努力是朝着完成的方向去的，有实际的价值，只是因为时间周期和资源的问题而导致了延期。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 程序员的生活就这样吗

我当了快十年程序员了，终于老得可以来谈谈程序员的生活是什么样子的了。

### 读书历程折射出来的生活状态

程序员的读书历程：x 语言入门→x 语言应用实践→x 语言高阶编程→x 语言的科学与艺术→编程之美→编程之道→编程之禅→颈椎病康复指南……

网上看到的，当时会心一笑，皆因似曾相识燕归来。

对我来讲，读的书大体沿着这个路线，我心戚戚焉。

每年检查身体时，报告上都有一条：轻微的颈椎强直。

我的爸妈经常在电话里叮嘱：坐得时间长了就起来活动下，别落下颈椎病。

### 程序员某方面很迟钝

看一则笑话：

“有个程序员兼擅修电脑，有一次被一个美女请去修电脑，那个破电脑怎么也修不好，这时美女在他的背上说：‘这个很不好修吧，不如咱们干点别的吧，要不多无聊。’那个程序员觉得自己的技术实力受到了莫大的挑战和侮辱，咬着牙说：‘我一定能修好！’”

这个笑话告诉我们，程序员呆萌的外表下隐藏着一颗敏感的心，对技术有一种近乎偏执的敝帚自珍，技术是他的立身之本，是其栓尊严的马桩，不容别人践踏。

其实说实话，我和爱人没生孩子之前生气的过程，基本上可以用下面这段代码表述：

```
do
{
    if (girl.accept (boy.guess ()))
```

```
{
    boy.saySomeHoneyedWords();
    if(girl.smileThroughTears())
    {
        break;
    }
}

}while(girl.eye.hasTears() && !girl.heart.isTired())
```

硬币有两面，程序员其实也有两极，可能在某些方面迟钝，但在工作中可能瞬间化身“犀利哥”。

## 别人对软件开发从业人员的印象

2009 年之前我在 X 秀园租房住，物业有个 T 师傅，人很好，就是到那时也没在单位分着房子，一家四口住在物业的一间不足 10 平方的平房里，两张上下铺占据了大部分的空间。尽管如此，他对女儿的学习还是给予了最大限度的支持：淘了一台电脑。可惜他的女儿主要是用电脑来聊 QQ，玩儿 QQ 空间，养宠物……

那台电脑的内存是 256MB 的，因为它是如此傲娇，加上 T 师傅的女儿又不懂得心疼它，什么炫丽玩儿什么，还嫌杀毒软件耽误事儿把它们给卸了，于是乎，这台电脑经常歇菜、罢工，T 师傅就得找人修。

因为修暖气拉家常的缘故，T 师傅知道我是写软件的，于是就拉我给他修电脑，用他的话说，搞软件的一定会修电脑，当然，我确实会修电脑，于是在一个月黑风高的夜晚，带上螺丝刀和 Deepin XP Lite 出发了。

电脑还能启动，就是开机时间很长，鼠标老没反应，我花了半个多小时才看到配置。对于这样的机器，256MB 的内存，我真是没有信心可以让它流畅运行，但是背对高低床上的几双眼睛，我只能强自镇定，用 Deepin XP 重装了系统，哇喔，居然比原来流畅多了耶。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

我给这台电脑安装了免费的金山毒霸，光盘里带的，还安装了输入法。然后我在一家人的千恩万谢声中回家了，回到家里我就到网上下载了 Deepin 2000 以备后患。



我的担心并不是多余，此后我隔三差五就要去 T 师傅家做客。说实话，我真想给他装个 DOS!

后来我不在 X 秀园住了，才摆脱了修电脑的噩梦。

但是，2010 年我搬到新房子里，楼上有个搞建筑分包的 X 师傅，因为孩子上学的原因在小区买了套房。为人很热情，见人不笑不说话，装修时找到我，觉得我阳台的墙砖好看，从那以后我们就认识了。

程序员的生活就这样吗.🐼

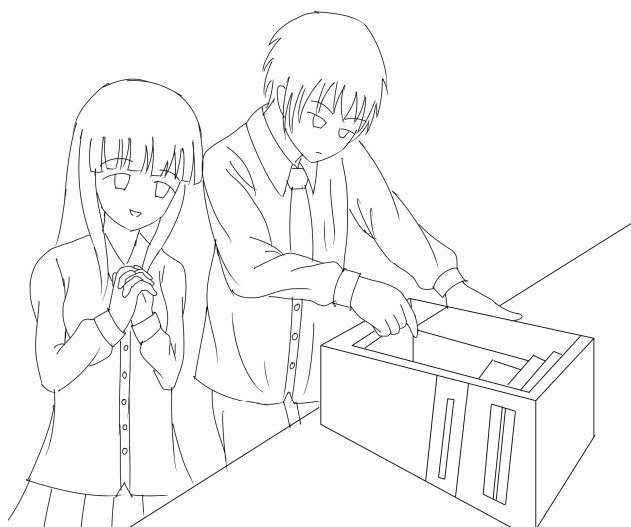
X 师傅儿子上初中，要考高中，目标是我们小区附近的西北大学附中。孩子爱玩儿电脑，X 师傅知道我是写软件的，然后就说：“这下可找对人啦，走走，给我看看电脑去！”

天哪，我怎么就是个修电脑的命！

我给 X 师傅修过电脑，搞过局域网布线，修过电信的 IPTV，还给他的 Android 手机安装过 APK……

你看，我什么都能干，因为我是程序员。

**猿哥哥修电脑好棒暖**



加班生涯不是梦……

我是程序员，我妻子搞测试。

有一阵子，大概半年多的时间吧，我基本上每个周六加班，工作日晚上加班，周日偶尔再加半天班。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

先看快下班时的 QQ 对话吧。

妻：今天加班吗？

我：应该不加吧。

妻：那几点回？

我：有点儿小问题，估计七点多吧。

妻：就知道你会这么说！

到大概七点半，我们又进行了电话沟通。

妻：回来了吗？

我：问题还没解决，快了。

妻：具体什么时候？

我：估计得到八点多了。

妻：等你吃饭不？

我：不用了，你们先吃。

等到快九点的时候，我们短信又聊上了。

妻：还回来不回来了？

我：马上回，真的，收拾东西了。

妻：别骗我！

我：真收拾东西了。

妻：问题搞定了？

我：差不多……

等过了十点半，妻子发来一条短信：门没反锁，我先睡了，你个骗子。



后来的后来，我终于告别疯狂的加班，可是妻子开始加班了，为期将近 9 个月。

当然，我一般不会问她什么时候回来，反正我自己和女儿玩得也挺高兴。

再后来我认识了楼上的邻居，老公是写软件的，老婆在华为，两口子加班比我们还厉害。

再后来认识了一哥们儿，也是搞软件的，上班地方离家十分钟脚程，他和妻子的对话，与我上面写的类似，经常是，六点说要下班了准备回家，到家有时九点有时十点有时不知道几点。我爱人这个时候终于理解了我：原来软件男都是这种对话逻辑。

## 领导说没看见你干啥

其实我们也不总是加班，有时候也有闲下来的时候，此时积极、热情、有追求的程序猿们会选择重构代码、优化性能等等在他们眼中很有意义的事情来做（必须说明，这些工作对软件



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

产品也是非常有意义的)。而这些事情往往是隐性的，它们带来的改善效果通常要过一段比较长的时间才能显现出来。这样一来，好多程序员在领导眼中变成了“无所事事”的闲人。

这几天一直都在做优化性能和用户体验，  
这些很重要。。。



## 明天新版本上线

我原来做的产品，支持在线更新。每次要释放新版本，都是一场噩梦。

所有的开发、测试、产品经理，甚至质管部门负责做基线的 MM，以及我们的老总，都会这样一个时刻，把心吊起来，大气儿都不敢出，又期待、又担忧、又希冀、又害怕地等待着最后一刻的到来。

老总可以在家睡觉，质管部可以说流程以后再补，产品经理可以说释放申请我们再说……可是作为开发人员，作为“程序猿”，我们可以说什么？总不能说，实在不行晚两天再发布吧。宣传文案都做好了，网站上都提前预告了，市场人员都和大客户说好了，这次版本要解决困扰大家已久的 X Bug。

可是，老板说了，不论多晚，都要给他打电话。市场说了，不管怎样，明天早上一定要给客户升级。项目经理说，拼了，要么见不到明天的太阳，要么明天太阳照常升起。你看这不是废话吗？关于明天的太阳，就这俩可能！

于是，我准备了烟灰缸；张三准备了沙琪玛；李四买了“闲趣”；测试 MM 很贴心，先把会客室的沙发擦干净，躺上去休息了，她们说：改好了叫我们。我们“程序猿”呢，其实也想去会客室的沙发上躺着啊。

到 10 点的时候，项目经理说，哥们儿，什么时候可以封版？不能再改了！于是大家忙着封版送测，测试经理也表示这个决定比较靠谱，谁说不是呢，没有 Bug 的软件不是好软件！只要不是一剑封喉就行了，小 Bug 无伤大雅，还能陶冶客户情操。

好吧，11 点的时候，全体开发人员都感到有些厌倦，项目经理说大家交叉测试，于是该我们等测试 MM 了。

项目经理说大家夜都熬上了，就把时间都用到刀刃上吧，都精神点儿，一会儿 Bug 就千树万树梨花开了。于是各位都去找测试 MM 聊天了。

凌晨四点的时候，大家双眼赤红面色灰败地聚在一起：一个从未发现过的超级 Bug 突然现身，一直扮演奥特曼的架构师表示不能完败这只怪虫。

怎么办呢？

有人提议告诉老板，延期上线。有的“程序猿”还不甘心，建议搞到早晨六点，还搞不定就回退代码或者上线昨天的版本。测试 MM 总结说，我们不认为可以在明天八点半之前上线版本。

项目经理说：早死早超生，今天不上线，改天上线还要熬通宵，你们以为跑得了嘛！

于是，“困兽”们咬牙切齿又奋斗了两个小时，奇迹出现了：谁也不知道那个 Bug 怎么被修复了，反正测试 MM 测不出来啦了，真是山重水复疑无路，柳暗花明又一村！

等到快八点的时候，项目经理拨通了产品经理的电话，产品经理拨通了技术副总的电话，

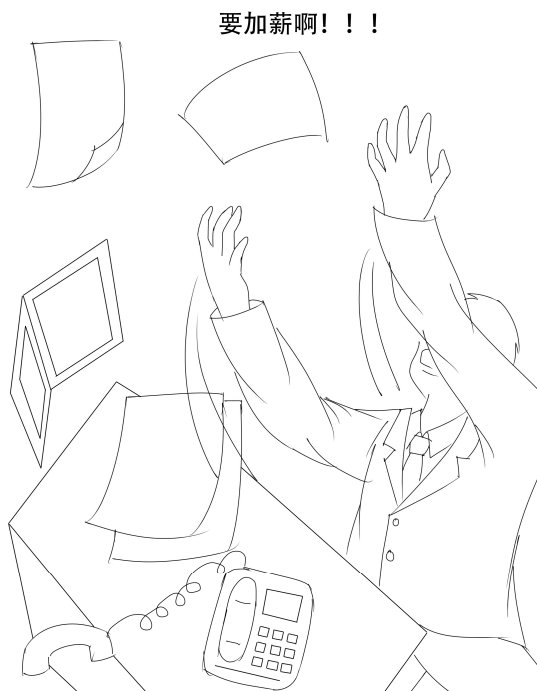
☎你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

技术副总拨通了老板的电话，老板拨通了市场部的电话，市场部拨通了客户的电话……于是，皆大欢喜的局面出现了，原本是白发三千丈缘愁似个长，结果成了柳暗花明又一村，三军过后尽开颜。

但，我的心中，总有一种欲说还休的忧愁，才下眉头，又上心头。

不出我所料，上午十点的时候，老板打电话给我说：赶紧的，客户那里出问题了！

想加薪的时候……



有时候你想加薪，考虑了许久，终于找领导谈话了。

“腾讯通”上的对话。

阿猿：领导，我想去做测试。

领导：为什么？我看你开发干得不错，最近那个 CMS 系统的推荐子系统，你做得非常好，跑了一个月没 Bug 啊。

阿猿：测试岗位工资比较高。

领导：兄弟，你的才干有目共睹，我早看到了。相信我，谁干活怎么样，我看得很清楚。我是无意间听说的。领导，我都两年多没涨过工资了，房价都翻番了。

领导：以后别再打听别人薪水了啊。

阿猿：最近生活压力很大，房东又涨租了，我还谈了个女朋友，工资一发，第二天就没了。

领导：行啊，没声响地就谈上了，啥时候见见？

阿猿：领导，压力很大啊。

领导：这年头，哪个男人身上不是几座大山呢。我一个月要还房贷，交托费，养车，给父母零花……

阿猿：领导，你看要是加薪水……

领导：放心吧阿猿，7 月份涨工资肯定少不了你，其实我已经跟 L 总说过这事儿了，他表示看好你，没问题。

阿猿：大概能涨多少？

领导：这得到时候看，现在我也说不准，不过一般 10% 吧，表现好的话能到 15%。

阿猿：10%，300 块啊，这……

领导：也可能 15%，如果今年公司效益好，还可能更多。我觉得呢，只要咱们把产品做好，加薪是必然的事情，你干得怎么样，我心中有数，我很客观的，也很看好你，觉得你潜力很大。

阿猿：我哥们儿介绍了个公司，谈差不多了，我想……

领导：那个公司给你开多少钱？

阿猿：也没多少，将近 6K 吧。

领导：有时候做事情还要看你能否成长，机会如何，你看咱们做的这个，将来必然火。

阿猿：生活压力太大，没办法，领导，那边让我周三给答复呢。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

领导：我找L总再说一下，争取给你加到5K，怎么样？咱们的产品还是很有前途的，你看看同行那几个公司，发展得多好。只要干对了方向，还怕将来宝马奔驰开不上么。相信我，我在这里干这么多年，也是看好产品的前途。要耐得住寂寞，量变会引发质变，不用多长时间，最多明年，产品就会大卖，到时鸡犬升天，少不了你的。

阿猿：要是能到5K，我还是愿意跟着领导干，其实我是挺愿意跟您干事儿的。

领导：那好，等我消息。

阿猿：好。

第二天，阿猿收到消息，8月份发工资，涨到5K。阿猿那个高兴啊，真真是剑外忽传收蓟北，初闻涕泪满衣裳。那萦绕在心头多日的阴霾，一刹间被风吹散了，煦热的光笼罩全身，每一个毛孔都说不出来的舒服，走路仿佛踩在云上。

到了8月份，工资发下来了，阿猿一看，一毛钱没涨！阿猿心头火起，当即垮着脸就找领导了，两人再次来到“茶话亭”。阿猿直截了当地问怎么回事。领导说大环境不好，国内经济持续下滑，至今未见反弹迹象，公司现金流也出现困难，大家这次都没有调薪，不过下次一定会的。阿猿说那我知道怎么办了。然后，两人都神情严肃地离开了茶话亭，阿猿去打电话，领导去电梯间抽烟。

过了一个月，领导离职了。阿猿被调到公司的另外一个开发部门，继续程序员的平凡之路。

## 软件开发的十八般乐趣

有人说我在传递负能量，看了我的博客（<http://blog.csdn.net/foruok>）上的“漫谈程序员系列”感到了前所未有的彷徨与迷惘。这不是我的初衷，我只是给你看真实的世界，真的勇士，敢于直面惨淡的人生与淋漓的鲜血。

总而言之，我们需要正能量，就像多数人喜欢大团圆结局的电影一样。上帝抛我们到这世上，给我们各种困顿、挣扎、忧思、恐惧，生活已经足够无奈了，何必再给人添堵呢。现在我们就掰着手指头来数一数，作为一个程序员，到底都有哪些欢乐。我要传递正能量！



### 憧憬带来的欢乐

我在“程序员零门槛吗”一节中，其实已经提到了一个问题：你为什么要做程序员？你

☞ 你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

的初衷是什么？别说那时我们年少懵懂，是被推到河沟里的。其实河沟里也有很多快乐的小虾米。

在我们开始选择时，不论是主动、被动，还是被前辈们坑蒙拐骗，其实都怀了对未来的美好憧憬。这种憧憬给过我们快乐，对一些人来讲可能很短，对一些人来讲可能很长，还有一些人可能觉得无所谓长也无所谓短——万能的三分法。

在开始的开始我们有憧憬的快乐，实际上在我们做产品的过程中，每一次新的开始，也都充满了阳光和期待，我们相信，每一天的太阳都是崭新的。我们也相信，明天太阳依旧升起，和煦的阳光还会照耀我们。

当你看尽人事苍茫，对什么都感到无趣时，心就萎缩了，幸福感也悄然离去。可以向往、追求的日子，无疑是幸福的，珍惜吧少年。

## 思维的乐趣



当程序员专注地思考如何更好更优雅地实现一段代码（设计、算法、解 Bug、预研……）

时，他的样子是最美丽最性感的，浑身都散发着柔和的光芒，他自己当然也能够从中体会到乐趣。

如果你给程序员的脑子里植入控制思维的纳米颗粒，通过智能手机之类的设备跟踪阿猿们的思维，一旦发现他不按照你的想法去想问题，就发出 **STOP** 的指令，那阿猿恐怕要像悟空被唐僧施咒时那样满地打滚了。**Google** 研究的纳米技术有望在未来实现这一点，感觉上比《一九八四》里真是高科技多了，到时出门都得装备天蚕丝织就的护身衣（据说这玩意儿刀枪不入也能防纳米冷枪），不然的话你可能瞬间就被人给控制了，想想就可怕吧。

因此，有形的枷锁不可怕，可怕的是不能学习知识，失去思维的能力。而程序员时时刻刻都在锻炼思维，这是一种无与伦比的乐趣。

## 掌握新技术时的快乐

很少有一个职业能像我们一样，每天都可以接触新的知识和技术，有那么多那么多的机会可以去学习，去体会，去应用。每当我们看到一门新的语言，一种新的技术，甚至一个具体而微的雕虫小技，都会发自内心地微笑。我们就像奔向三棵大橡树后的姥姥家的小红帽一样，看什么都是新鲜的。

喜新厌旧是人的天性，而软件开发给了我们喜新厌旧的理由，只要我愿意，随时可以应用新技术新框架，我们可以堂而皇之地以性能、设计、灵活性等理由来学习、实战新技术，这种得天独厚的条件，还有哪个行当可比呢？

“吾生也有涯，而知也无涯，以有涯随无涯，殆已”。

这是《庄子·养生主》里的原话，有人推演庄子的意思：我庄子的生命是有限的，但我面对的知识是无限的，要我以本来有限的生命，去追求那永远看不到尽头的知识。

而我是这么理解的：生命有限，知识无限，以有限的生命来追求无限的知识，直到生命的尽头。简单说，就是：活到老学到老。还有一句话，可以作为注脚：朝闻道，夕死可矣。



你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 挑战极限的快乐



如果你看过《蜘蛛侠》，一定对那个双手会吐丝的家伙很感兴趣，他白天羞涩黯淡，面对邻家女孩总不能顺畅地表达他的爱意，可是到了晚上，他却成为不朽的传奇。他在高楼大厦之间呼啸而过，他可以拉住要冲出悬崖断轨的火车。现实中也有蜘蛛侠，奥地利的迈克·凯米特是徒手攀岩的高手，他用十分钟就攀上了 180 米的高楼。

我们程序员，也像蜘蛛侠一样，看起来有点儿木有点儿呆，其实内心有一股不灭的火焰，挑战技术极限会让很多人爽到爆！

挑战难题，是很多程序员刷存在感的方式，他们每天不干别的，就在等待这样的机会。寻寻觅觅寻不到，山重水复疑无路，为伊消得人憔悴，谁知道峰回路转，柳暗花明又一村，蓦然回首，百万秦关终属楚！

那一刻，此处省略 500 字……

```
while(1) programmer.smile();
```

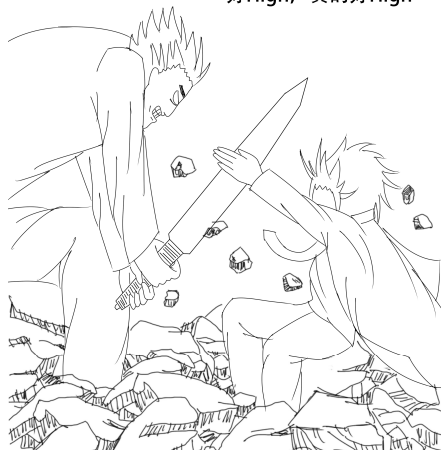
## 好为人师，助人为乐

作为“程序猿”，有很多机会能够让你一展指导者的风采。从小我们就被教育要助人为乐，当时我们不理解什么是助人为乐，现在我们可以深切体会到这一点。当你用你的知识和技能帮助别人解决问题时，当你把你的知识和经验分享给别人时，你会感到由衷的快乐。

假如你有一百元钱，分五十给别人，你就只剩下五十了。假如你有一瓶子知识，你把他们全部分给别人，不仅你的那瓶子还在，甚至反过来你还会得到一些（这是常有的事儿，所谓共鸣和强化，碰撞和升华也）。知识和技能就像快乐一样，你越是愿意分享，你就越是能得到更多。它们比物质的那些东西强太多了，而程序员恰恰就拥有这些神奇的宝贝。

## 与高手同行

好High，真的好High



☞ 你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

我很喜欢温大写的《神州奇侠》系列，萧秋水和李沉舟都是非常激荡人心的角色，十年前看过，二十年后不忘。

男人在成长的过程中，总是看着自己前面的那个男人的背影长大的。程序员也一样。我们需要一个标杆，优秀的程序员渴望与高手同行，哪怕只是远远的一瞥，也是令人兴奋的。没有高手相伴的世界是寂寞而不可想象的。

程序员的世界里，就像侠客的江湖一样，各种高手比比皆是。

## 修复 Bug 的快乐

生病时才觉得平常有多好，假如能给我再来一次的机会……

程序员的世界里，这样的机会很多很多，那就是：修复 Bug。

你看着一个人遍体鳞伤，你希望治好她，然后，你做到了，不需要任何嘉奖，有机会做又能做到，就是最大的快乐。

## 埋地雷的恶趣味

太邪恶了吧，有的程序员喜欢故意在自己的软件里埋雷，看看究竟能“炸翻”谁。



## 你爽到了，我也爽到了

程序员的职责其实是满足用户的需求，解决用户的痛点。如果有一个客户用了你写的软件，觉得非常棒，他们藏在心底不好言说的需求都被你照顾到了，那他们就会对你的工作给予至高的评价：见人就说这玩意儿不错，愿意向朋友们推荐你的产品。甚至有的用户会联系到公司、个人，非要主动说一声：你们的软件太牛了。

此时此刻，也许你表面上还有点谦逊，内心恐怕已经乐开了花，转过身来可能就泪流满面。高山流水遇知音啊。

## 简单生活的快乐

不知道你有没有一种感觉：想简单很困难。

有些人觉得交十方朋友，磨万千关系是一种快乐，游刃有余乐在其中。

有些人则天生羞涩内敛，爱在心头口难开。我就是这种人，所以我还是热爱技术，这是一种选择，我想也不算是逃避。该做的事我们还是会去做，该走的路一步也不会少。只是在有些时候，我们可以活在一个简单的世界里，不用考虑那么多，你说 **Hello**，他就回应你 **Hi**，这是多么好的体验。

作为程序员，我们希望能把精力用在正确的、值得的事情上，痛恨内耗、消耗、被消耗。

我们可以！就这么简单。

当你不需要应付纷繁芜杂凌乱成麻的各种关系，能够随心所欲地把精力释放到你愿意的事情上时，就会是快乐的。专注做事，收获简单而纯粹的快乐，这是上天赐给程序员的礼物。

## 功成名就

王江民开创了程序员的传奇时代，那想必是有厚重而绵长的快乐的。

江湖很大，大成者少，小成者众，不管大乘小乘，不落下乘，都是汗水之后的结晶，快

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

乐是自然而然的事，也许伴随着泪水，也许伴随着一口恶气，也许伴随着轰然倒塌的身躯，也许平平淡淡于无声处听惊雷。

虽然大侠传奇的时代已经远去，然而时间永是流逝，江湖依旧多梦，侠少们仍然有理由鲜衣怒马呼啸来去，寻梦之旅随时都在开始。



那么，让我们用一句话来结束我们的快乐之旅：软件开发中不是缺少快乐，而是缺少感知快乐的心灵。

## 你的幸运女神呢

每个人的身边都有牛人。也许这句话不大准确，应该这么说：每个人的身边都有比自己牛的人。

那么，问题来了：为什么别人比我们牛呢？为什么别人运气好呢？

### 气死我啦

要么是别人工资比我们拿得多得多。

要么是看着和自己差不多，结果人家有豪宅有豪车。

要么是你怎么都觉得那人比你还不如，实际上人家在单位外面有自己的公司和事业。

要么是你看着哪儿都不如你的那谁谁，几年不见，居然干到主管了，而你还在一线苦苦地敲代码。

要么是你觉得旁边那位，整天不见干活，怎么领导就是有好事儿就找他。

要么是你老在加班，拿钱比你多的那个人却很潇洒，一下班拍拍屁股就走了。

……

所谓人比人，气死人，就是如此吧，距离你太远的那个同行，可能很牛，牛到天上了，但你不会在意。我们在意的，其实是身边的人为什么比我们“强”。注意，我给强字加了个引号。

为什么人心难平？因为比较。

为什么会这么比较？因为人都觉得自己不应该过得比别人差，所以就想看看这是不是真的。

为什么人都觉得自己不该过得比别人差？

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 为什么我没有机会

我们看到别人平步青云，日子风生水起，常常会顾影自怜：为什么我没他那种机会？

机会这种事儿，其实很玄乎。

有些人成功了，会说：没什么秘诀，只是运气好。

千万别相信这种鬼话，没有谁比谁运气好，运气面前人人平等。我们之所以感觉某人运气好，那都是人家努力经营的结果，很少有平白无故遇上好运气的事儿。

我要说的是：人人都有自己的机会和际遇，就看你能不能抓得住。

我们常常听人讲：哎呀要是当年我怎么怎么着，现在就不至于怎么怎么着了。这就是没抓住机会导致的后悔。

也有的女人会抱怨：怎么我当年就跟你这个程序员，要不然早成富太太了！

我们这一众程序员看到这里该多伤心啊：跟了程序员就成不了富太太……

其实不是这样。互联网这个世界里一夜暴富的例子太多了，就看你上了哪条船。我们要讨论程序员的事情，他们的女朋友或者拮据的太太咱就先不提了，后面再让她们亮相吧。

摘录一段张小娴的散文《漂泊，但快乐》里的话：

“不是我们去选择际遇，而是际遇选择我们。当际遇来到，我们能够做的，是接受这个挑战。不要说机会从来没有出现，它曾经出现过的，只是你舍不得放下自己拥有的东西。”

事实多半如此：际遇在我们面前时往往我们不觉得它是际遇，或者因为种种原因我们没有接受这个际遇。

## 程序员怎样才能有好运气

行文到现在，貌似还是没有破题。

题目是：你的幸运女神呢？

答案是：机遇只给有准备的人。

就这么简单。

那么问题来了：怎样才算是有准备？

未来要自己创造，你自己的努力与自强，是转运的起点。

## 实力与运气

这个世界上几乎所有事情的成败都取决于实力和运气两种因素，但很多人经常把二者搞混或弄错，更不清楚它们各自占多大比重。实际上实力就是确定性的部分，是运用技能解决问题的能力；而运气则是不确定性的体现，是不可控的偶然因素。

它可能阴差阳错降临到某个地方、某个人头上，但这个人能不能应对、驾驭、享受好运气带来的山乡巨变，则依赖于这个人的实力。

互联网上有句话广为传播：站在风口上，猪都能飞起来。这句话讲的也是运气。

可是还有一句话：风停了，摔死的都是猪。这句话讲的是实力。

## 程序员如何看待实力与运气

程序员所处的行业和世界，外面看起来风云变幻，但是对程序员本身来讲，其实确定性的因素更多：技术立身，技术本身更具确定性；靠技术吃饭的人，更倾向于脚踏实地追求稳定。

对于程序员来讲，到底有没有运气这事儿？

其实我觉得更多时候，我们把现状的不同归结于运气的时候，往往是因为自我感觉自己的实力与比自己过得好的人不相上下。但我们往往忘了，实力并不仅仅是技术能力，技术能力只是实力的一部分，有了技术能力你就有了起飞的基础，但还需要你主动地去追求、创造、



把握自己的机会。

朱买臣是汉朝初年会稽人，很喜欢念书，对其他事情都没什么兴趣，和有的程序员只喜欢写代码类似。朱买臣家里十分穷苦，每天到山上砍些木柴，挑到市场上去卖，勉强过日子。不过朱买臣觉得，自己只要勤勉上进，终有一天会时来运转，能够一偿读书报国的志愿。

朱买臣天天念书，挑柴的时候嘴里还不断背诵诗文，就这样努力到四十多岁，还是苦哈哈的，看起来没有任何要发迹的样子，他的妻子终于忍受不了了，闹着要离婚，整天大哭大闹，朱买臣只好答应离婚。

转眼之间，又过了几年，朱买臣快五十岁了。有一次，朱买臣听说会稽郡有一位太守要到长安去，就想办法以运卒的名义跟了去。到了长安以后，朱买臣抓住一个机会，见到了汉武帝。汉武帝很赏识他在《春秋》、《楚辞》方面的见解，派他做中大夫。后来因为他提供了一个对付东越的报告，汉武帝一高兴，就派他回会稽当太守，让他衣锦还乡。

当年被人瞧不起的樵夫，摇身一变为太守，乡民们夹道欢迎。他的前妻又羞又恼又怒又悔，在请求朱买臣再收留她未果后，寻个机会上吊自尽了。

朱买臣的故事，说明一个人只要肯上进、肯努力，一定会有所成就。但这里面其实还有非常关键的一点，你要识别机会，主动抓机会。如果朱买臣不是主动跟着会稽太守去长安，不是主动找机会撞见汉武帝，那后来的际遇也不会出现。

对于程序员，其实是一样的。勤勉努力提高技术实力是一方面，另一方面，还要培养自己的软实力，你要了解自己所处的行业，了解所在的城市的业态，了解与自己技术相关的公司、产品的动态，同时用这些东西再反哺我们的技术，这样当环境有所变化时，我们才能捕捉到自己的机会。这是讲我们要动态地看待能力，主动地观察机会。还有另一面，身边的人会带给我们机会。因此程序员除了打怪练级搞技术，还要多和身边的人沟通，你能干还要和更能干的人耳鬓厮磨，让别人知道你能干，并且愿意干出点事儿来，这样当别人的际遇到来要乘风起舞时，那你的机会可能也就来了。

## 打造自己的专属名片



前面我们说程序员要修炼内功，还要经营关系，这都比较抽象，这里我们再具体一些，来谈谈程序员的名片。

假如你要印一张自己的名片，除了名字之外，只允许你写三句话。你准备写什么？

或者换一个问题：假如你的简历只能有三句话，并且不超过 30 个字，你准备写什么？

这个问题的核心是：找到最能描述你自己的词语，展现你的核心竞争力。

每个人都有自己擅长的一面，你可以用它来代表自己。有些程序员还有知名的产品，可以用它来给自己正名。有些程序员是知名公司的创始人，你当然也可以在名片上印相应的头衔。更多的普通程序员，实际上是个手艺人，在软件开发的世界里，作为一个工具被人用来锻造各种产品。我这么说绝无贬义，这只是现实的一种，当你还不是君子时，只能是器具（参

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

考“君子不器”的解释)。那么作为器具，一定有自己的独特功用，这是程序员要想明白的事儿。比如感冒药有千百种，但个个功效不同，你一定要找到你的独特疗效。

我们接受自己是个“工具”这个事实，然后就要努力寻找自己独特的作用，给自己一个标签，容易让人记住的标签，持续地去经营这个标签，这样时间久了，你接触过的人都会知道你是个什么样的人，擅长什么，能干什么。最终当某个机会到来时，抓住机会的人就会在熟识的人群中划拉，看看张三能干什么，李四能干什么，一个萝卜一个坑儿，坑坑都填平，大家伙儿一起来干事儿，把事儿干成。

程序员在别人心目中是什么形象，90%是由自己决定的。你表现出什么，别人就了解什么。假如你对图像处理算法特别熟悉，写出了美图秀秀，那别人一定会知道你是美图秀秀的作者，你擅长图像处理；假如你精通视频算法，自己写出来解码 H.265 的 `decoder` 并且贡献给了 `ffmpeg`，那别人也一定会知道；假如你擅长团队管理和激励，你带的队伍士气高涨，屡屡攻城拔寨，那别人一定也会知道这人是个将才，带队伍打仗很靠谱……如果一个程序员留给别人的印象就是“这家伙就是个混日子的主儿”，那我估计，他还得想办法混下去。总之，你需要做的，是找出自己擅长的方向，实实在在地做出点儿事儿，树立你自己的独特形象，这样，机会降临到你身上的概率才会大一些。

## 那些害死程序员的细节

这次，我们不谈权术，不谈潜规则，不谈技术，不谈理想，咱们就来说说那些害死人不偿命的工作细节。

### 姗姗来迟



迟到是件小事儿，是吧。

很多人都这么想，我上学时也浑不在意这个问题，80%的课连去都不去。

可这真是大事儿，天大的事儿。

## ☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

迟到，首先反映的是你的态度，你对这件事儿不重视，所以你才会迟到。这是态度问题，态度决定一切，所以：

- 如果你与女朋友约会时迟到，面临的是一场气。
- 如果你面试时迟到，承受的 99% 是被拒绝。
- 如果你上班时迟到，除了被扣钱，还会强迫领导给你贴一个“这厮不自律，啥都不在乎，不靠谱”的标签。
- 如果你开会时迟到，可能被迫买冰棍给大家吃，还可能遭遇他人的白眼。
- 如果你拜见客户时迟到，很可能会丢单。

.....

总之，守时是一个程序员的基本工作素养。什么，你老加班？那是另外一回事儿……

## 请假不提前

这个还要特意说，真有必要吗？

真真是很有必要。

项目经理、部门经理、总监之类的角色，我不幸都干过。员工请假这种事儿几乎天天都会发生。有的人会发个短信，说我那啥啥有个啥事儿，明天不来了。有的人会丢给你一封邮件，说那啥啥明天圣诞节陪女朋友逛个街。有的人两天没见影儿，来了跟你说领导我有点急事儿去见了个网友……总之各种情况都有，只有想不到，没有遇不见的。

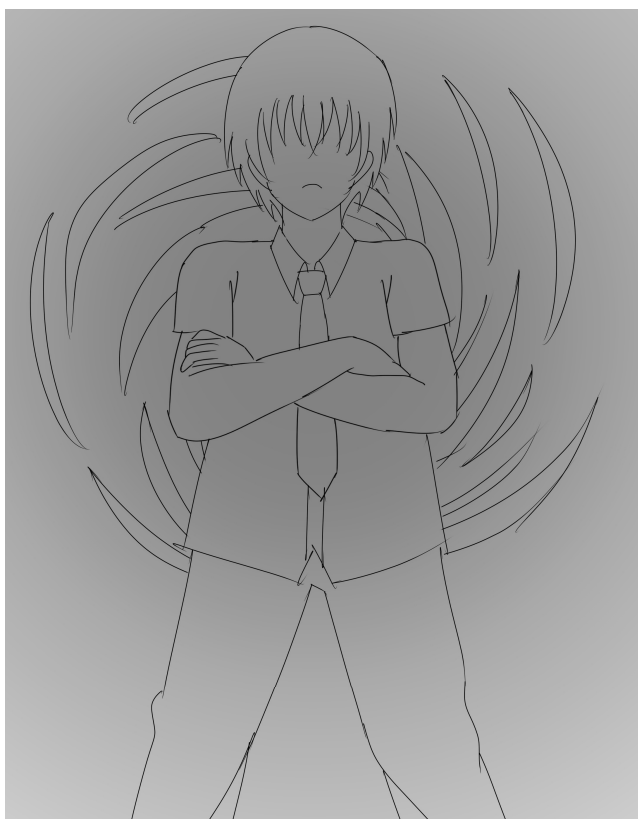
“请假”这个词儿，两层含义：请示、给假。先请示，后给假。

你一条短信，一封邮件，一个 QQ 留言，这是通知，是先斩后奏。如果你连这种通知也省略了，玩儿的是说走就走的潇洒，那你的领导对你可就没好脸色了。

有的程序员觉得工作这么忙，加班还不一定搞得完，面对面不好意思跟领导说。完全不必这样，真有事儿了谁挡着你呢，大家都是人，领导也是人，将心比心么，谁没个头疼发热，谁没个三朋两友，谁没个老婆孩子，谁没个爹妈……人有那么多关系，总会有事儿发生，只要你安排好工作，通情达理的领导不会无情到拒绝的。如果你仅仅是告知，更甚之先斩后奏，

那领导就会有被绑架的感觉，有什么后果可想而知。

我是黑洞



不少人觉得我们程序员不善言辞，这不重要，别人说的话，随便听一听。

但是，不善言辞并不意味着我们可以成为黑洞：吞噬一切，从不反馈。

反馈状态与结果，这是程序员必须要重视的、非常非常重要的一个习惯。

我们拿到手的每一个任务，都是面向结果的，有时间盒限制的。难道有哪些公司给你一

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

个任务时会给你说，随便搞搞，无所谓？

我见过相当一部分程序员，领导安排工作任务时他不说话，默默地接受。好吧，虽然我期望你来和我讨论一下这个任务的具体情况，可你不在意我也只能认为你接受了这个任务预期的结果和时间期限。可是，有的程序员不这么想，或者至少他的行为在暗示别人他根本不在意这事儿做成什么样子。真的，有些人从不反馈自己的工作状态。任务完成了就撂一边去，遇到问题了就放在那里，自己搞点别的事情，比如打会儿游戏，刷刷网页找些段子乐呵一下，看看视频娱乐娱乐，可就是不跟相关的同事说，不跟领导说。

这样做的后果是很严重的，你的同事们的工作可能依赖你，你的领导可能在等着你出结果好安排下面的事情，可是你恍若无事地把自己打造成了一个“黑洞”。当同事问起，当领导问起，想想吧，各种吐槽：“你应该告诉我一声”、“怎么不早说！”、“为什么不说！”、“真耽误事儿！”。什么反应都会有的。兄弟，你的形象，那就一落千丈、江河日下了。久而久之，你在大家心目中就不再是值得信赖的队友，你这么做是主动逼迫别人不信任你，接踵而至的必然是领导对你的微管理。

正确的工作习惯是及时反馈，紧急的工作优先反馈，让你的伙伴们知道你的状态，这样大家才能相互配合一起前进。

## 不知道对结果负责

举个例子吧。

领导让张三组织开周会，要求大家都参加。到了周五下午三点开会时，领导发现李四、王五、赵六以及孙二娘都没到场，就问张三。张三说，哦，我通知他们了，他们说有事儿来不了。

这就是一种典型的面向过程的做法。

你做了吗，做了。

你达到预期结果了吗，管它呢！

“你让我做，我做了，至于结果，和我没关系。”

这是不成熟的表现，我们要去做一件事，当然不仅仅是做了，而是要做出结果，做到位，这才是应该追求的。

我自己也常犯这种错误，我给自己的说法是：尽人事听天命。其实都是错误的。你可以在失意时这么安慰自己，但绝不能拿这种态度和习惯来对待工作。

## 让别人给你当保姆

怎么说呢，还是举个例子吧。

领导给了小三一个任务，让她在 10 天内建立一个讨论 PHP 的 QQ 群，群成员要达到 250 个。

过了半天，小三找领导说，我不知道怎么建群，你帮帮我嘛。于是领导帮她建了个群。

第二天，小三找领导说，领导，帮帮我啦，我不知道 PHP 是什么。领导和小三讲了半天，小三似懂非懂地走了。

第三天，小三找领导说，领导，还是不行啊，PHP 程序员到哪里找啊，你再帮帮人家嘛。领导说你到 PHPChina、PHP100 这样的地方去找就行了。快下班的时候，小三又来找领导，说，领导呀，我研究了一天也没找到注册账号的地方，还是你帮我找找吧。领导打开网页，指了指右上角，说你看登录、注册就在这里么。小三笑了，哎呀领导你真厉害。

第四天，小三找领导说，领导，你帮我琢磨一下，我该怎么样发这个帖子呢，写成什么样子那些程序员才会对我们的群感兴趣呢？

第五天，小三找领导说，领导，都五天了，才拉了一个人进群，怎么办呢，有没有什么好办法，教教我。

.....

OK，看到这里我想你也烦透了。

这里面有一个很吊诡的情形：看起来小三是“发展 PHP QQ 群”这个任务的所有者，可实际上她通过不断地抛问题给领导，把这个任务推给了领导去完成。



你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

这是一个非常经典的管理问题：猴子在谁背上？

这个问题不但管理者要想明白，我们普通程序员也要想明白。我们自己的工作任务，是要自己想办法去解决的，要走心，别人可以帮你分析给你指点，但不可能把你的事儿帮你完成。要是你每走一步都攀着别人，迟早无人可攀。不管你是无心还是有意的，都不要这么做。

这不是我的问题

结果，到底是谁的问题？



这种事儿天天都在发生……

测试人员提了 Bug，张三说是李四的，李四说是王五的，王五说这是需求的。领导说服务器

吊死了，所有用户不能登录，项目经理就说这一定是那个小二黑抛了个异常，小二黑说其实是因为火工头陀代码不规范……

我要说的是，作为程序员，遇到问题，应该先接纳、解决问题，如果不是自己的，也要协调组织相关的人一起来解决。拼命地解释自己的理由、客观原因或者推说是别人的过错，这会给领导形成“此人善于搪塞、推诿、逃避责任”的感觉，以后真有好事儿时，你也没机会了。至于那些被你拉来垫背的同事，哼哼，走着瞧吧。

## 抢功

演艺界有很多“戏霸”，不管剧情发展，不吝剧情需要，总要突出自己，既搅了别人的戏，又破坏了整个电影或电视剧。

程序员的工作环境虽然简单，但也有一些不出奇的哥们儿姐妹儿，不在正经路上下功夫，专瞅机会抢别人的功劳。比方说一个任务两个人或三个人共同完成，验收时他一张嘴，没几句话，就把别人完成的那部分归拢到自己头上了。这是同事之间，也有领导抢下属功劳的。项目成功了，向老板汇报时，就是自己多么努力多么有计划有策略，完全不提团队成员的辛劳与汗水。更有甚者，有些领导凡事儿都推给下属，成功了就把各种文档、结果一整理划归自己名下，失败了就把各种原因一划拉都扣到下属头上。

遇见这种事儿，搁谁身上都气愤不是？

有相当一部分程序员是内敛型的，被抢功了腹诽一下然后默默忍受。有的程序员眼里不揉沙子，会选择机会以牙还牙。有的程序员不愿意争，这种事儿多了最后就自己离开这个伤心地了。还有的程序员会找领导澄清事实。当然也有人发动大家伙利用群体的力量来对付这些抢功的家伙。

不管怎么样，作为一个团队，如果总有那么两个人爱揽功爱抢功，就会把团队的气氛搞砸。而作为团队的 **Leader**，如果不能把功劳分给同事，不能把责任担在自己肩上，那这个团队就更难有凝聚力了。

所以，我讨厌抢功的人！

你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 打小报告

那个谁在公司做私活，还有那个谁和谁谁办公室恋情，有个女员工我很看不惯她，还有那个谁上班时聊天，有那个谁天天准点就下班



我讨厌打小报告的家伙，不解释！



## 背后拆台

有人的地方就有江湖，有江湖的地方就有八卦。有些八卦娱乐大众，无伤大雅。而有些八卦，其实是伤人的。

是非皆从口出，如果树欲静而风不止，有人非要拉着你骂骂那谁谁，你听听就算了，别跟着起哄哄，转过身也许他找别人说道你去了。

🗣️ 你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 千奇百怪的程序员

干开发时间长了，遇见好多好玩儿的程序员。

看看你躺枪了没。

留一手

神代码，柯南也搞不明白啊



有个哥们儿，在一家合资公司做程序员，能力挺强，写代码很神的。

他的代码我看过，深得 MFC 的精髓：大量的宏，很多函数就两三个宏，一个 BEGIN\_XXX，

一个 `END_XXX`，中间再夹另外一个宏。诺，这就是汉堡包了。

我碰巧看他的代码（题外话，没和他在一个公司一块儿待过），那个头大啊……这玩意儿也能记得住嘛，厉害，真是厉害。

就是这么厉害个人儿，他的代码只有他能看懂，他的地盘他做主，没人敢进去改东西。

这都不是真正的厉害之处，最厉害的是，这哥们儿代码里有不少 **Bug**，每个版本都不改完，故意留上两个，要是项目经理急了，就把旧的改完，埋两个新雷进去。

因为没人动得了他的代码，他就一直这么做，项目经理恨得牙根儿痒痒，可愣是没办法。

## 全局变量控

看一段代码：

```
/* Exported variables */

char hfile_name[MAXARGLEN+1], location[MAXARGLEN*2], *ppm = NULL,
    *gif = NULL, tzfile[MAXARGLEN+1], units[MAXARGLEN+1],
    units_abbrev[MAXARGLEN+1], tadjust_tzname[MAXARGLEN+1];
char next_ht_text[20], next_ht_date[20], next_lt_text[20],
    next_lt_date[20], *geometry = NULL, loadunits[80];
char *fgrise_color_arg = NULL, *fgfall_color_arg = NULL,
    *fgtext_color_arg = NULL, *fgmark_color_arg = NULL,
    *fgmllw_color_arg = NULL, *fgmiddle_color_arg = NULL;
int Usetadjust=0, Itadjust = 0, tadjust = 0, utc = 0, list = 0,
checkyear = 0, text = 0, skinny = 0,
    now = 0, graphmode = 0, htttimeoff = 0, ltttimeoff = 0, tstep = 180,
    middle = 0, mark = 0, mllw = 0, lines = 1, PPMWIDTH = 960,
```

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

```
PPMHEIGHT = 300, hinc = 0, tinc = 0, loctz = 0, iscurrent = 0,
curonly = 0, topline = 0, hincmagic = 0, calendar = 0, banner
= 0,
weekday = 0, hairy = 0, linegraph = 0, ps = 0, noampm = 0, utc
= 0,
have_offsets = 0, Ihttimeoff = 0, Ilttimeoff = 0, newload = 0,
subproc=1,OnlyTCD=0, datemdy=0, mapZoom=0;
double llevelmult = 1.0, hlevelmult = 1.0, Illevelmult = 1.0,
Ihlevelmult = 1.0;
time_t next_ht = 0, prev_ht = 0, next_ht_adj = 0,
prev_ht_adj = 0, faketime = 0, epoch = 0, mark_time_adj = 0;
double amplitude = 0.0, htleveloff = 0.0, ltleveloff = 0.0, DATUM,
marklev,
absmax = 0.0, absmin = 0.0, fakedatum = 0.0, fakeamplitude = 0.0,
Ihtleveloff = 0.0, Iltleveloff = 0.0;
double next_ht_amplitude, next_lt_amplitude; //mgh+
int num_epochs = 0, first_year = 0;
```

恐怖吗？这是我在一个项目的一个源文件里看到的，还有很多类似的源文件。

话说，有些程序员就好这口儿，特别喜欢全局变量，只是，代码写完的第二天，他就可能被自己的全局变量们给吞噬了。

## 别想看我的代码

还有的兄弟，对于自己的代码，非常之在意，他把自己的代码当作圣姑了，虽然必须行走江湖，但是我就是要给它戴个斗笠蒙个面纱，就是不让你看。有时候你看他一直埋头电脑前，噼里啪啦的击键之声不绝，可是就是不知道他写了什么。他从来不让别人看他的代码，谁看跟谁急。不知道他是怎么想的，我有时候只能猜测，他啪啪啪敲键盘，其实是在聊 QQ，他之所以不让人看，完全是因为，半个月过去了，一行代码还没写呢。当然还有一种可能，

他的代码比较凌乱，不想让人看出来。我们也可以往好的方面猜猜，此君自视甚高，认为自己的代码高深莫测，怕别人看了自惭形秽，怕打击别人柔弱的小心脏；或者此君认为自己的代码太优秀了，怕别人偷学了去。

不管怎么样，死捂着代码不放，终归不是个办法。

## 祥林嫂



“祥林嫂”原本是鲁迅小说《祝福》中的人物，是旧中国农村劳动妇女的典型。而“祥林嫂”这个词儿本身，现在又有了特别的含义，我只取其“牢骚满腹”、“不断 goto 当初”之意。

有个别程序员，对自己的技能比较珍视，采用二分法，认为这些技术是好的，是灵丹妙



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

药，认为那些技术是坏的，是穿肠毒药。尤其是产品技术方案没有选择他认为好的那个技术时，他会有很长一段时间不能接受，但凡项目执行过程中遇到点儿不顺，就会唠叨：“要是当初选择 XXX，肯定没这些问题”。类似的话，会不断从他嘴里说出来，循环往复，一咏三叹，简直让人崩溃。

## 我能干，我不干，我很忙

也许你遇见过这样的队友：分解任务时，说到哪个模块，都能头头是道，看起来没有他不懂的，可是你让他干时，哪个都不愿意干，老推说我已经 **take** 了多少多少 **task**。到最后，他基本上没什么 **task**，有限的几个还是很简单的那种，但当你跟踪进度时，他却总是说遇到了这个问题，还得多长时间，遇到了那个问题，又得多长时间，你要帮他看看时，他又说不用，自己能搞定。

## 未来战士

未来战士们对新的技术和工具特别热衷，每个领域的前沿技术他都知道，都能说出个一二三，而且这些技术更迭时他也都知道，比办公室里的其他人都知道得早。对于那些新技术，一旦他和你说起来，就神采飞扬，唾沫星星子四溅，此时即便你有事儿都不好意思走开。

未来战士所开发的软件所使用的 **SDK**、工具，只要有新的版本出现，一定要更新。哪怕是昨天刚刚更新过，今天也要更新。你不让他更新他就坐立不安，一定要来说服你，新版本有千般好，旧版本性能差、漏洞多、使用不便、编译缓慢等问题都会被找出来。

未来战士的哲学思想非常明确：新的就是好的。

## 石器时代的守护神

与未来战士相反，另有一些程序员，对自己掌握的技术以及对既有产品所采用的技术坚定不移地维护，你要想让他把软件依赖的框架版本升级一下，那是比登天还难：**Run** 得好好的，干嘛要升级？用现在的版本可以应付一切！

石器时代的守护神具有很强的韧性和对抗一切新技术的定力，只要旧系统还可以用，他就可以旷日持久地坚持、维护下去，因为他们的原则就是：只要将就能用，不到万不得已一定不要去更新，因为任何新的技术新的改动新的设计新的重构都是怪兽，不知道会带来多大的破坏。

## 老子不需要帮助

见过没？有的程序员就爱自己研究，就算是我被一个问题绊住，吭哧吭哧搞上一星期还搞不出来，你也别想帮助我！他们的信条是：老子能搞定，老子不需要帮助。

这种类型的程序员特别敏感，特别害怕别人瞧不起他，当然有时也有些自卑、多疑，担心别人的帮助里夹带着嘲笑，他们的敏感、多疑、骄傲、自卑等复杂情感驱使他们拒绝接受别人的帮助，也驱使他们忘记时间盒的存在。

## 快来帮帮我吧



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

嘿嘿，这种类型的程序员，最爱说的就是：哎呀搞不定了，谁来帮帮我。

只要你一帮他，这活儿就成你的了，他们会让你觉得帮人不到底你好意思么你，还会让你觉得送佛不到西天就是你的罪过。我的天，他们遇到问题都不想的，直接拉你过来调代码。

他们是“老子不需要帮助”的反面。

## 别让我读代码，我是文盲

我懂得一门语言的所有语法，关键字认得，控制结构懂得，继承知道，多态了解，设计模式我也清楚，可你别让我看代码，看不懂，真心不懂。

这也是一类程序员，他们能够了解编程语言的所有语法和细节，可就是看不懂别人的代码。

## 精雕细琢的艺术家

咦，这里该不该断行？

代码编辑器里，到底这种字体好看还是那种字体好看？

这里的条件判断，到底是三层 IF 语句嵌套好还是三个 IF-RETURN 好？

……

这些艺术家们把代码当作一门艺术，从编辑器开始打磨，一直到每一行代码到底多少个字符合适，花括号是放在行尾还是行首……都要仔细斟酌研究。对他们来讲，美观是高于一切的。

## 孤独的人



有些人选择软件开发，就是因为：再也不用和别人打交道了。

所以，他们会回避任何与别人交流的机会，谈话、电话、邮件、即时通信、微信……总之，他们理想的工作环境里只有代码没有人类。与同事或客户沟通，对他们来讲是巨大的负担，感觉说一句话就会耗费他们一辈子力气。

你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 咦，你也在混日子啊



戳你一下，疼吗？

### 混日子的定义

来自百度百科的定义：“生活等方面过得不怎么好，无目标，混混沌沌”。

混日子：“即没有理想，没有抱负，糊里糊涂地生活。也指工作没有责任心，得过且过”。

那么，如何界定某一段日子是不是混过去的呢？

首先，我们在时间轴上选取起点和终点。

其次，回顾起点处自己的状态（包含技能、精神、灵力等等），回顾终点处的状态，比较这两个状态，看看有没有哪方面有进步。

假如任何一个维度都没有令人欣喜的变化（排除自我要求过高的情况），那基本上就可

噢，你也在混日子啊.🐼

以说下面这句台词儿了：

“这段时间都不知道怎么混过去的”。

## 瞎胡混呗

拿程序员来讲，有时候是忙一阵松一阵。有项目了很忙，没项目了很闲。闲的时候就容易混，反正领导也没派什么活儿，反正过一阵儿还得忙，混过这阵子再说。看起来是合理的啊，老那么忙，老加班到后半夜，谁受得了呢。

那忙起来呢？其实忙又分两种情况，有目标的忙和没方向的瞎忙活。瞎忙活也是混日子的一种。看起来脚不沾地，一会儿这儿一会儿那儿，实际上干的事儿对自己都没什么意义，这就等同于混日子。当然这是被动的混日子。也有主动混日子的瞎忙活，就是“让自己看起来很忙”，对吧，其实我没啥事儿干，也不想干什么事儿，又担心被领导发现我闲得要死，只好装装样子，让自己忙活起来。这是主动混日子。

混日子的合理性就在于：你没告诉我要干啥，也没告诉我干成啥样。

其实是这样的：**自己没有目标，加上被安排惯了，一旦没人安排他了，就只好混日子了。**

## 天天混的程序员

这个问题我好像提了很多遍，不过还是得提：“阿猿”，你为什么干开发呢？

为了可观的薪水？

为了享受创造的快乐？

为了自我成就与自我实现？

工作其实有双重意义：

- 挣钱养家养自己
- 自我价值的体现

## ☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

但实际上，有一部分程序员看的是第一个，或者干着干着，工作的意义就只剩下每月的那点儿薪水了。这样的情况，是最容易导致混日子的。

有些程序员的一天是这么度过的：

- 9:00 到公司。
- 9:00~9:30 ，吃着买来的放心早餐（包子、煎饼果子、面包等等），等待电脑启动。
- 9:30~10:00 ，浏览网站，看看新闻，关注一下 NBA 或者英超。
- 10:00~10:30 ，哎呀，今天该干点啥呢？想着想着把页游打开了。
- 10:30~11:00 ，还没怎么玩呢，就十一点了啊。
- 11:00~ ，收下邮件吧，看看有没有什么工作上的事儿。
- 11:30~ ，吃什么呢，开始和小伙伴们商量中午吃什么。
- ~13:30 ，关了游戏网页，小睡一会儿。
- ~14:30 ，被别人叫醒了，据说有个 Bug 要处理。
- 14:30~15:00 ，抽根烟去，困死了。
- 15:00~16:30 ，打游戏，抽空浏览 JIRA ，抽空给测试 MM 说两句，就说我正在研究呢。
- 16:30~17:30 ，切换到 JIRA 页面，开始看 Bug 描述，告诉测试 MM 描述不清晰，无法重现，让测试 MM 再测，重现了叫他。
- 17:30 ，出去抽了根烟，回到座位上又玩了几分钟游戏，准备下班。

特别说明：游戏只是占位符，浏览网页只是占位符，新闻只是占位符，英超只是占位符，抽烟也是占位符……可以被其他形式替换，比如找人聊聊电视剧，上个淘宝，类似的，都可以。

严正声明：看我博客（<http://blog.csdn.net/foruok>）不能用于替换占位符，我传递的都是正能量，磨刀不误砍柴工，刀磨快了干什么都好使。

日子，一天天地就这么过……

咦，你也在混日子啊！



## 混日子的感觉

我在刚参加工作时，经常会觉得自己在混日子，因为每天都在等待，等待设备出事儿，然而设备并不是分分钟都出事儿，所以就有一些时间要混过去，上网聊天，同事们聚一块打麻将，日子就这么过去了。这种感觉很不好，大概是因为我虽然是个懵懂的家伙，却不是一个能心安理得混日子的人，所以混起来就有一点惶恐，一点惭愧，一点不安，一点迷惘，一点不知今夕何年的感觉。

我不喜欢这样，感觉自己就像一块地，没人来耕种，野草野兔小鼯鼠都很 **HAPPY**，我很荒芜。快乐是即兴的，转瞬即逝的，怅然若失的忧愁却是如影随形的。

为什么没事儿干还能拿薪水你还不快乐？

那天看到一句话：不快乐是因为可以像只猪一样懒，却无法像只猪一样懒得心安理得。

原来如彼！



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

关键词是：**心安**。

我心不安宁，所以我不快乐。

不知道你混日子的时候是什么感觉？

## 干掉混日子这个魔头

如果你混日子混得心安理得，就别往下看了。

## 为什么混日子

为什么混日子？这也是一个问题。

我接触过一些程序员，表现出来的行为让人怀疑他们的想法和小学生类似：写程序是给公司写的，干活儿是给公司干的。总而言之，这就不是我的事儿，我把八小时交给你就成了。

这也是部分程序员干活不走心的一个源头。

不过这说到底还得赖教育，下面的话你是不是经常听到：

- 给我好好上课
- 给我好好做作业
- 给我好好吃饭
- 给我好好玩儿
- 别给我看电视

……

就是这样，从小我们都被教育坏了。有的人能把这种错乱纠正过来，有的人就这么一直迷惑着、迷糊着，成人了也还觉得是为别人而活，干的工作是给别人干的，谈的女朋友是给别人谈的……

其实，我们都是为自己而活不管你是被安排任务的“码农”，还是给别人安排任务的项目经理，还是被人遗忘在角落里没活儿可干的替补“码字猴”，日子，都是你的，你要为你自己的日子负责。不想荒废，那就往下接着看吧。

咦，你也在混日子啊.🐼

## 找一个目标

要想杜绝混日子，就得正本清源，搞明白为什么会混日子。

其实混日子的定义里已经说了：缺乏目标。

目标决定行动，找一个目标是绝顶重要的事儿。这个目标还得你愿意相信。然后，就朝着这个目标努力吧。



在向着目标前行的过程中，也许你会感到劳累、辛酸和疲惫，不过，一切都是值得的，因为你要的幸福，就在那片更蓝的天空，只要你的翅膀卷起风暴，挣脱自我的牢笼，你就可以像狂风一样舞蹈，飞得更高。

你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 让程序员无语的那些事儿

听说嫁人要嫁程序员，钱多话少死得早。这话多半是程序员自己“黑”自己的。程序员是有非常特别的幽默感的一群人，善于自嘲，勇于自“黑”，耐受力超强，很多事无可无不可，不到是可忍孰不可忍不会冲冠一怒。不过，就是这么随和的人，也有一些受不了的事儿。

### 需求变化

这个，这个，嗯，就加这几个简单功能吧



为什么把“需求变化”排第一呢？

因为有人说：杀一个程序员不需要用枪，改三次需求就可以了。

由此可见，需求变化的杀伤力有多强。

我见过不少程序员，和产品经理（需求人员）关系紧张，话不投机半句多，或者关公秦琼比脸红。在程序员看来，往往需求人员一句话，程序员几天的工作都白费了。程序员是最珍视自己劳动的，心里能不窝火吗？你早说嘛，有变化你早说嘛，为什么非等我干个差不多了才说呢，你早说我能不让你改嘛……

## 加班

加班呢，在程序员的字典里，是个热词。不折不扣的热词。

很多程序员过的是“固定程式”般的生活——上班、加班、回家开夜车然后再起床上班，周而复始。

因为大家都知道有加班这么一回事儿存在，其实，一开始领导要求加班，虽然我是不愿意加班的，也还是从了。赶进度，一起联调，市场时间点定了，都情有可原。

可是，我真的不愿意加班，对于无情可原的加班，我是深恶痛绝的。

比如，明明一个项目需要 20 个人月，老板一句话，4 个人干俩月就要交活，项目经理居然还答应了，这明摆着一开始就把加班算计在内了，让高智商的程序员情何以堪。

比如，明明是张三的活儿出了岔子进度没控制住，却把我李四拉过来加班搞定，我白天的尽职尽责不就是为了能够按时完工不加班么，张三白天吊儿郎当看新闻、打游戏、聊 QQ 却让我来收拾这烂摊子，我能平衡吗？

比如，需求讨论了几遍了，都确定了，程序员都没日没夜干了两个月要发版本了，领导一句话，说哪哪不合适，产品经理一句话，说哪哪不符合用户习惯，需求一句话，说哪哪还要改一下……你一句话不当紧，你还说“这个应该很简单吧”，你不知道的事儿多了，我们的程序就像蛛网，彼此纠缠，关系那是剪不断理还乱，牵一发动全身，你这一改动，说起来简单，实现起来可能要从网页一直改到数据库还有手机客户端，那是全线崩溃，结果呢，不加班能行吗，我能加得开心吗？

比如，老板说，我们产品做不好，就是因为这帮开发人员老不加班，时间投入不够，于是，我们要加班。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

比如，线上系统周六出个 Bug，找张三张三不在，就拉李四来加班，现趴代码现找问题。

## 很简单

最怕外行领导内行。因为他们不能理解技术细节，总是从外面去看程序和软件，经常会对程序员说，“这个应该很简单吧”。

要知道，“这很简单啊”，这话是多么惹人恼怒。比如你让张三实现一个登录功能，你认为就是在界面上放两个输入框，一个用户名，一个密码，再加上一个确定按钮，能有多复杂？其实呢，你看到的只是冰山一角。从程序员的角度来看，用户名是否合法，密码多少位合理，与后台通过什么协议交互，后台如何保存这些信息，要不要支持多点登录，用户权限是否很多是否要根据用户角色来分配，支不支持找回密码，登录过程中网络异常了怎么办，来电话了又需要怎么处理……你看，程序员要做的工作，是不是比你想象的要多？是不是一堆一堆的细节需要考虑？这还是说得出来的，说不出的，还有一大波与登录相关的 Bug 正在赶来……

所以，“很简单”这种话，不应该从需求、项目经理、老板等嘴里蹦出来，你们不是程序员，不了解程序员的痛有多深。就算你曾经是程序员，可是你也不写程序很多年，早不能原音重现了。

作为一个程序员，我特别讨厌“这个很简单吧”这种话，可是又经常遇到，甚至我自己有时也会说，但这话真的让人很无语，除了显示说话者的无知、自以为是、炫耀之外，再没别的什么好内涵了。

## 角色错位

项目经理干程序员的活儿，老板干项目经理的活儿，程序员操心老板的活儿……

有一部分小公司，角色就这么乱。

出现这种现象，一般的过程是这样的：

项目经理是程序员出身（码而优则仕），技术很牛，虽然当了项目经理，可还没从“我

自己可以搞定一切”的自慰自大中跳脱出来，一旦项目进度赶不上预期或者某个程序员的进度出现问题，项目经理就会把别人背上的猴子抢过来，重新下沉到写代码或解问题的细节中，一旦项目经理这样做了，更坏的事情就接二连三地发生了。因为如果项目经理被具体的问题绊住，无暇顾及团队管理和项目管理，项目的执行情况就会越来越离谱，进展堪忧，可项目经理可能还存在幻想，觉得等我搞定这几个问题再回头来收拾局面也来得及。这真是童话啊。

老板是不信童话的，他会看到项目经理的问题，心里也火烧火燎的，于是就伸出手来，帮着项目经理管项目，这个人闲了，那个人没事儿干了，你应该先做这个功能，你们老不开会怎么能了解团队状况，各种各样的担忧和实际看到的问题，会乱了老板的心，让他身不由己地插手项目执行层面上的事儿。好了，现在老板下水了。

程序员是清醒的，或者自认为清醒的，他们能看到老板的越级管理，直觉上就认为这不是老板该操心的事儿，他们会觉得，老板应该放手，应该去考虑更高层面的事情，那老板到底应该干什么呢？这个问题会攫住一部分程序员的心，还有一部分程序员反感项目经理的做法，认为项目经理不信任自己，也怀疑是不是项目经理心里看不起自己才插手自己的代码。

你看，就这么错位了。

一旦错位局面形成，要扭转就需要九牛二虎之力。

首先要改变的是技术出身的项目经理，他认为技术是立身之本，舍不得丢弃，担心自己没了技术以后不好发展，他一时半会儿看不到投身项目管理所能产生的效果，这需要有人告诉他、帮助他、陪着他走过这段吊诡的岁月。另外老板也需要捆住自己的手，给团队时间，让团队自己去解决问题，如果你觉得项目经理还没合格，那你可以给他找个教练，但不要自己下场比赛。

## 亲，能不打断我吗

当程序员陷入潭水深深深几许的思考中时，一个电话，一个提问，一声“进展如何”，甚至一句“吃饭去吧”，都会像一记大招一样将程序员丰富的内心世界轰得粉碎。相信大家都会体会过自己睡得正香被人喊起或吵醒的愤怒，没错，就是这样，高速行驶的途中就怕这些状

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

况，哪怕是一只小鸟迎着你的玻璃窗撞过来，也可能导致交通事故。

## 改别人的 Bug

女儿小的时候，拉臭粑粑后我会给她擦屁股，没觉得臭也没觉得有什么恶心。可如果让我候在一个大男人身边，等他拉完屎了替他擦屁股，那我无论如何不能接受！

你的代码就是你的孩子，无论他丑还是美，遇到问题，你不会追究石榴树上为什么结出樱桃，你义不容辞地去 Debug 它，而且因为是你亲生的，你一般不会嫌弃。

别人的代码，别人的 Bug，那就是另外一回事儿了，听见“你帮张三改个 Bug”这类的话都可能产生生理反应，真没办法，顶不住领导施加的压力，短衣襟小打扮撸撸袖子，不情不愿地进了张三的属地，也很可能收不到好效果。原来，这里只有一条通往“蜘蛛巢”的小径，张三已忘了，李四寻寻觅觅寻不到，结果误入代码深处。

## 维护老代码

老代码啊老代码，可能已经被许多人维护过了，可能有“拿个锤子看什么都是钉子”的程序员到此一游，可能有“爱贴膏药”的程序员到此一游，可能有“随地大小便”的程序员到此一游，可能有“瀑布式”程序员到此一游……

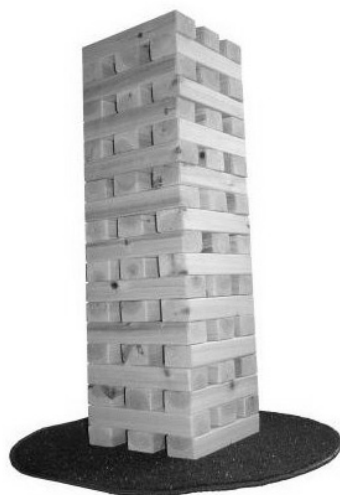
在老代码里，你可能看到很多硬塞进去的、突兀的、生硬的、硌人的代码，这是“拿个锤子看什么都是钉子”的程序员们的杰作：不管某个功能放到某个类、某个模块里是否合适，硬钉进去完事儿。

你也可能看到代码里到处都是“膏药”，比如调用某个函数可能有溢出，“爱贴膏药”的哥们儿就在调用的地方添加各种纠偏和预防措施，不管合不合适，这会儿不出错就行了。

你也可能看到有些标注为修复某个 Bug 的代码，用的都是临时解决办法，而这些临时的代码，居然还隐藏了很多一触即发的 Bug。

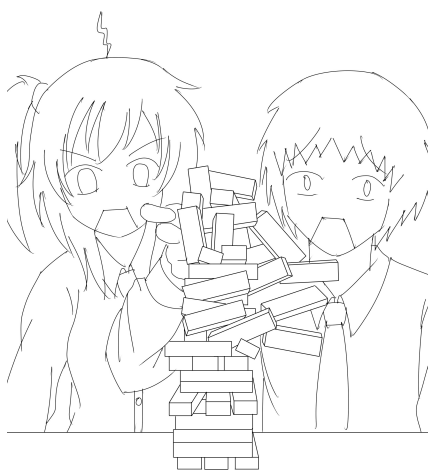
你还可能看到有人留下的“超级大瀑布，一个函数超过 2000 行，飞流直下三千尺，十次翻页看不完……”

老板一直认为你们的系统是这样的：



看起来健壮、层次清晰、结构紧凑、易于扩展，所以，老板觉得加个功能满足一下客户的小需求很简单。

老板不知道，你和你的小伙伴们都要哭了：





## 多任务的神话

我使用 **Chrome** 浏览器，一般会同时打开多个网页，每个网页占用一个标签页。当网页数量小于 8，我能清楚地从标签页上看到网站 **icon** 和网页标题。当网页数量超过 12，网页标题基本就看不到了，每一个标签都会缩得很短。当网页数量超过 20 个时，网站 **icon** 也看不清楚了。当网页数量超过 40 个时，**Chrome** 就会变慢了，有时甚至会崩溃。有一阵子我的 **Chrome** 总是不停地崩溃，我以为电脑出了问题，但后来发现是我打开的网页数量实在太多。

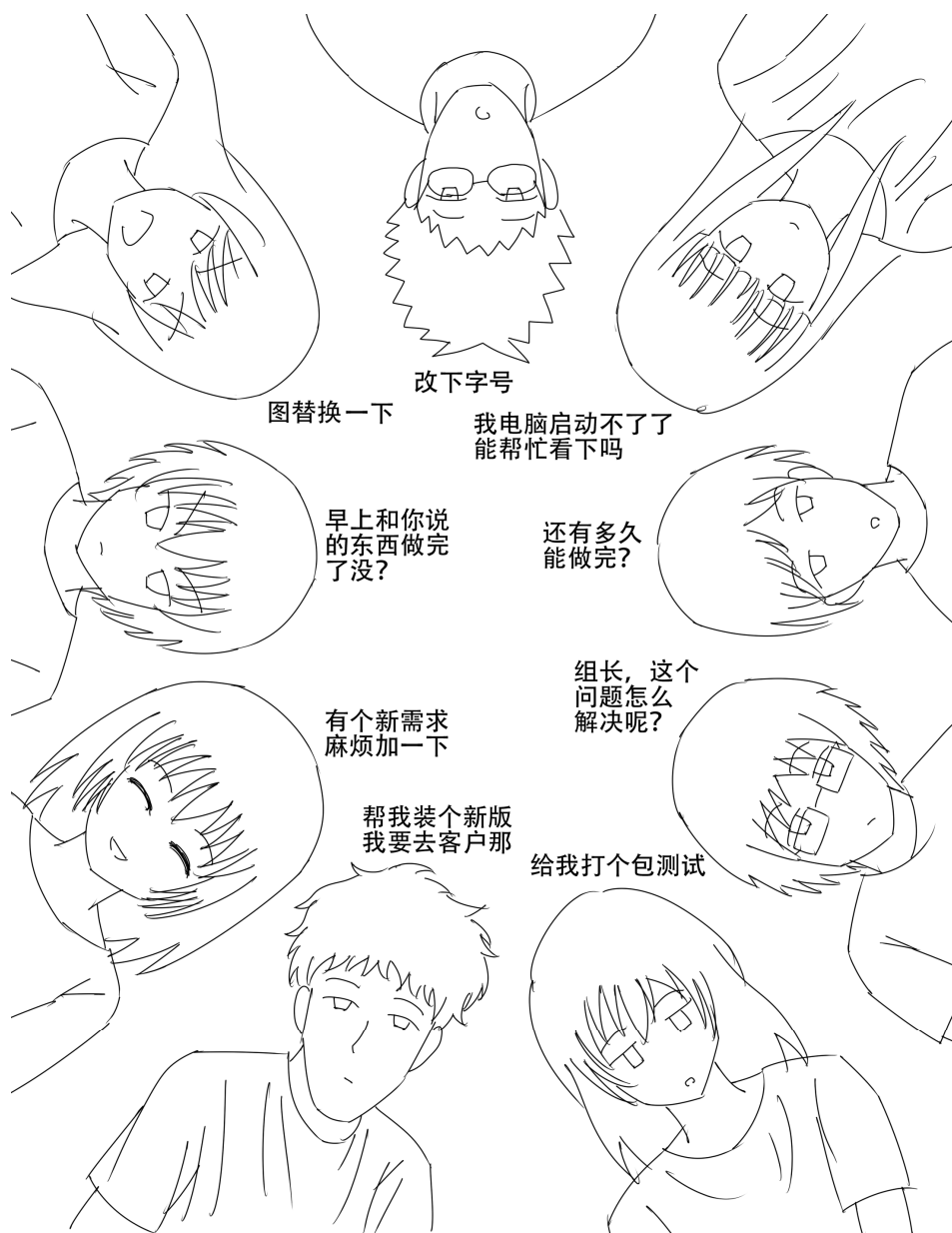
我们通常以为，浏览器可以同时打开很多网页，每个网页都可以并行实时加载，这样做能够大大降低我们的等待时间，能够显著提高效率。但实际情况并非如此。

一方面每一个标签页会对应一个 **Chrome** 子进程，网页数量过多，内存、CPU 等系统资源占用就会变多，操作系统花在调度不同任务上的时间也会变多，不但 **Chrome** 会慢下来，整个电脑都会慢下来。

另一方面，过多的标签页导致看不到网站 **icon** 和网页标题，我们在切换网页时，要么需要把鼠标悬停在标签页上等待 **Chrome** 给出的提示，要么需要直接点开某个标签页来查看，总之我们需要花额外的时间来确认某个标签页对应的是哪个网站。这是我们花在任务切换上的开销，时间久了开销会很大。

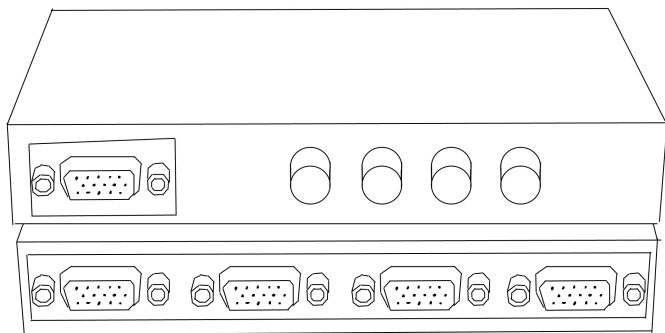
这里拿浏览器做例子，只是想说明，同一时间内干多件事情，不见得效率会更好，恰恰相反，效率会因为多任务同时进行而急剧下降。

在 **BOSS** 的眼里，很多时候程序员就是万能的 **Chrome** 浏览器，他们下意识地认为，只要把任务一股脑儿地丢给程序员，程序员就能多头并进，高效地完成它们。甚至有的 **BOSS** 认为，一个任务需要五天，如果你见缝插针，两个工作量差不多的任务，可能只需要六天。



BOSS 会在我们做一件事的时候，突然插进来另外一个任务，比如你正在开发新版本，老版本上爆出一个 Bug，你就需要修复它，可能你花了一天甚至更多的时间才解决这个 Bug，此时你原来正在做的事情的交付期却到了，BOSS 就会来找你要东西，要是你拿不出来，BOSS 就会很奇怪，“怎么回事儿，不是说得好好的这周五提测吗”，就好像他根本不知道中间你被抽走做别的事情他的。当然你可以提醒他，然后他会说，“哦，那再多给你一天时间，这样没问题了吧”，你可能这么想——“别的任务花了我整整三天，你居然只给我一天”，我认同你的看法，不但如此，我还觉得 BOSS 应该给你四天时间，因为你要重拾中断的思绪，要知道，千丝万缕很难一一捋顺，效率和产出都需要缓慢爬坡。

可是，BOSS 会选择性忽略这一点，别人有时也会。他们认为程序员身上有一排多任务按钮，就像用于多主机共享显示器的切换器一样，只要你按下按钮，瞬间切换，就没有额外消耗。



很多人都认为你身上带了这种多任务切换按钮，或者他们虽然没这么认为但就是循着这个思路做的。比如你正在写程序，你的同事会叫你帮他看个 Bug，或者让你收下邮件，或者让你参与测试用例的评审，或者问你格式化代码的快捷键是什么，或者问你知不知道某个新出的技术是怎么回事儿……这是零零碎碎的小任务干扰，我们不得不响应，我们在群体之中，各种沟通渠道都要畅通，看起来这样才能相互配合，高效地工作。但实际上，过多的干扰会大大降低我们的工作效率，我在“就这几点，提高工作效率”一节中做过详细分析，这里我不再多说。

队友之间的沟通和交互是必需的，如果你阻塞与外界的所有沟通渠道，就会把自己变成

一个孤岛，就会削弱团队的力量，如果人人都成为一个孤岛，那就没有团队了。所以，我们能做的，就是找到一个平衡点，在个人效率有少许下降的情况下，通过协作，提升团队整体生产率。

2014 年年底，我负责为一个初创公司组建研发团队。接近年底，大部分人都不太愿意动弹，都想着年后再说吧。我从 12 月份开始招人，一直到 2015 年 1 月份，也没敲定几个人。我几度下定决心自己来写代码，先做移动端 APP，再写后台，预计有组建团队的几个月时间，我自己要是专心开发，也可以做出一个基础版本，可以开展小范围的产品测试了。我真的尝试这么做了，可是真的静不下心来，我脑子里整天都盘旋着去找哪个熟人来给我介绍开发人员，我该怎样和应聘者沟通，该如何让他们对我所在的初创公司建立信任（这个真的有些难），我还会考虑、测试各个招聘途径对于初创公司的有效性，如果我要找以前的同事来，还要跨越一些心理门槛经过不少内心挣扎。有时你谈好的人又会出现意外情况，比如他被老板说服或者他突然想到了什么事儿不愿意来了。总之仅仅招人这一件事，就会带来很多很多情况，这些或预料中的或突发的状况，三不五时地降临到我身上，虽然不是时时刻刻都有状况，但一个打不通的电话、一个“我决定不去你们公司”的 QQ 留言，就可以影响我一天的心情。

在这种变化纷繁意外不断的时段，往往你刚打开 Android Studio，还没写两行代码，注意力就一下子破裂了，就像太阳底下多彩的肥皂泡那样，再想重新聚焦，非常之难。所以我在春节前那两个多月的时间里，既没有招到人，也没有写多少代码。这种结果真让人沮丧，对我不小的打击，内心的焦虑与日俱增。后来春节实在太近，只剩下三周左右，我绝望地意识到，不论我怎么卖力地演出，也吸引不来几个观众了，于是我真就死了心，暂时把招聘一事搁置起来，埋头写起代码来。经过两三天的焦点转换和效率爬坡，终于驶入了快速开发的高速路，不到两个星期，就完成了 Android 平台 APP 的基础版本。我的信心终于重建，我以为过完年再花上两周就可以 Beta 测试了。

春节之后返岗，面对求职找工作的黄金季节，我重拾了“没有研发团队绝对是不行的，招人是头等大事”这种意识，于是春节前短暂的开发工作再度搁浅，我重新陷入“求之不得，寤寐思服。悠哉悠哉，辗转反侧”的微妙挣扎中。

从 2014 年 2 月底，一直到 2015 年 4 月份，我的 Android APP 再无一丝进展，因为，我又被招聘这件事给攫住了。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

啰里啰嗦这么久，我只是想说，多任务真的不是能让开发人员高产出的银弹，它只会大大降低程序员的效率和团队的生产率，“ $5+3=10$ ”这种匪夷所思的事儿会不断因为相互纠结的多任务而降临。因为即便你在一个程序员的工作负荷中多增加一个项目，他的时间损失也会超过 20%，如果你让一个程序员同时处理三个项目，那他超过 50%的时间就都浪费在任务切换上了。

程序员决不是安装了任务切换按钮的“神兽”，他们实际上是在表演戏剧，每一幕的切换都需要重新布置场景，比如暮景、凳子、桌子、凉亭，甚至演员还要换妆，这都需要时间，因此，如果能让程序员在一个时间段内（一周、两周、一个月……）聚焦在同一件事情上演独幕剧，就会大大提高产出，这是项目管理人员和 BOSS 们应该也必须首先考虑的。

# 乐观的程序员

( 写于 2015-04 )

程序员办公桌必备：

通宵神器：泡面，功效HP+100

加班神器：咖啡，功效CON+100



有一次看到来自 Instagram 用户 findheart 的话：

程序员是一个懂得享受生活的群体。每几个月或者几周，他们就会庆祝一个叫“Deadline Eve”的节日。在这一天晚上准备大量的食品饮料，通宵达旦地工作，并互相致以“下个 milestone 绝对不再拖延”的美好祝愿。

我从这个笑中含泪的小段子看到了程序员的一个特点：**乐观**。

程序员是我见过的最乐观的一群人，这也是我作为一个程序员回顾自己的开发生涯时发

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

现的一个秘密。

要知道，程序员的生活充满未知：未知的项目，未知的语言，未知的框架，未知的工具，未知的进度，未知的天坑，未知的 Bug，未知的交付日期，未知的绩效评估，未知的 KPI……

在这种茫然四顾一切皆不可知的环境中生活，你要是不乐观，绝对扛不住，你根本不可能见到明天的太阳。因为，未知会产生恐惧，这是人之天性，恐惧堆积多了人就会崩溃。而伟大的程序员，坚韧的程序员，不可思议的程序员，拥有超强耐受力的程序员，倍受蹂躏仍踽踽前行的程序员，不但奇迹般地在这个非人的环境里存活了下来，还每天乐乐呵呵屁颠屁颠的，这不是乐观是什么？还能有哪个群体如此乐观吗？

## 看，项目估算就这么简单

当程序员在白色的电脑屏幕上写下“`printf( " Hello World! " )`”时，他确信，只要按下 CTRL+R 或者单击运行按钮，黑黑的命令行窗口就会对他们说“Hello World!”。

是的，就是这个样子。当你写下“Hello World”这句骇人听闻的咒语后，电脑决不会用“Goodbye World……”这句更震撼人心的话来回应你。

我们程序员面对的世界是“0、1”世界，简单而直接。我们在软件开发的世界里徜徉得越久，就越容易受到这种逻辑思维潜移默化的影响。我们崇尚简单，讨厌复杂，无论是在数字世界，还是在现实生活中。

程序员喜欢看到确定的、可预知的结果，而我们的世界里却有这么多的不确定性，我们的办法就是只往好处看，只青睐那些我们把握得住的东西。

:Begin //请注意，此处的 Begin 为用于 goto 跳转的标签

当我们被迫评估项目进度时，我们会先采取大事化小小事化了的策略，然后把那些人见人爱的小家伙——拎出来，乐不滋滋地给他们分时间，这个 4 小时，那个 3.5 小时……，很快我们就觉得胸有成竹，一切尽在掌握，拎着我的“巴拉拉魔仙棒”傲娇地转上一圈再伸臂一点，整个项目大概需要 4 个人月，预留 1 个人月应对随时可能出现的天外来客，我们有 8 个人，2 周半搞定！

大家伙分头行动，哼哧吭哧咣咣咣开始了。

第一周的时候，我们报告说：“工作正在有序进行，一切都在掌握之中。”

第二周的时候，我们报告说：“出了点儿小意外，但加两个晚上班儿就可以追回进度，我们可以搞定。”

第三周的时候，我们报告说：“又出了点儿小意外，你知道，软件开发就是这样，惊喜不断。不过我们周六和周日已经摆平了意外，我们很快就可以搞定了。”

第四周的时候，我们报告说：“嗯，情况比我们想象的复杂，在集成时遇到了不明状况，整个团队搞了一个通宵才搞定，结果第二天没办法干活，所以，进度比预期的要晚一些，不过，请相信我们，很快就可以搞定了。”

### 通宵连续加班第N天 ……





☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

第五周的时候，我们报告说：“这周肯定可以搞定了，相信我们的能力。”

第六周的时候，整个团队都一脸“黑线”在加班。

第七周的时候，我们报告说：“系统集成完成啦，第一个 Beta 版本可以内测了。这虽然比我们预期的时间晚了一点点，不过，我们的内测版本堪称完美，几乎不可能出现 Bug。”

.....

第十周的时候，整个团队都在加班加点改 Bug，人人一脸“黑线”外加两个青肿的眼袋，每当焦虑万分的老板询问上线时间时，项目经理都会说“快了快了，就这两天就搞定了。”

等到第十四周要结束的时候，版本终于发布了。

皆大欢喜啊。

老板已经被这群猴子逼疯了，心里恨得要冒火，还得咬着牙说：“兄弟们辛苦了，你们的辛苦大家有目共睹。”

我们报告说：我们发布的版本质量很高，容易上手，用着也方便，用户一定会满意。相比之下，少许的延期是值得的。

*goto Begin*

好吧，项目估算简单吗？

“一切管理理论都是扯淡，哪一种也不适用于我们的团队，因为我们这些程序员是独一无二的，都有着与众不同的个性，我们必须找到自己的项目管理方法。并且，我们现在的团队已经在我们的独特的管理模式下运转良好，完全不必要引入学习成本那么高的控制手段，况且那样对大家也是一种刺激，觉得公司不信任大家，这会大大地降低士气。毫无疑问，我们现在的团队士气高昂，必须要保持……”

这可能是某个项目经理说的话，我不记得了。他最终的结论是：“虽然这次项目 Delay 了，但是下次，绝对不会了！我们一定有办法控制进度。”

看到了吧，这就是我们的乐观。它生动地告诉我们什么是真正的乐观：即便无数次实践

已经证明我们的估算和控制手段不靠谱儿，我们还是会信誓旦旦地在下一个项目来临时许下“绝对不会延期”的诺言。

## 这很简单，两天搞定

程序员群体的乐观是由个体的乐观汇聚而成的。

有多少次我们被问到这个问题需要多久解决时，我们都说出类似“两天搞定”的话。

有多少次我们看着同事深陷泥沼无法自拔完工无望时，我们说出“要是我早就搞定啦”之类的话。

有多少次我们要学习未知的技术并把未知的技术应用到未知的业务上最终交付给客户一个可用的系统时，我们都会天真地从我们既往的经验池里抓几枚硬币来撒撒，然后说出“大概需要半个月”这样的结果。只有程序员，彪悍的程序员，无所畏惧的程序员，敢于乐观地面对未知的一切拍拍脑袋给出一个确定的工期。

.....

这都是因为我们程序员倾向于把事情想得简单，在我们这些崇尚简单的程序员眼里，即便你秉持着不悛以最低的效率来评估的原则，也会给出一个最终看来还是乐观的估计。

这让人绝望吗？

不，恰恰相反，这正是我们程序员之所以为程序员的一个关键特征，惟其如此不能活也！

## 没什么难的，不过如此嘛

万事开头难。

很多事情只要我们突破了最初的障碍，后面就会顺风顺水势如破竹。对程序员来讲，这种规律很容易被强化，因为，我们有太多机会学习新技术、新框架、新业务，而大多数的语言和框架，写一个万能的“Hello World”就入门了。

这种学习新技术的感觉，和开车类似。我学会开车后，一直没有上路跑过，提车后犹豫

## ☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

了几天，总觉得城市里路况复杂，很多车不管红绿灯总在强抢黄三秒，行人又乱闯，电动摩托总是突然呼啸而过，左转右转与直行同时被允许，想多了总觉得自己不能应付这种状况，就在考虑要不要请个陪练陪着我上路跑几天。后来因为着急上牌，没办法就直接上路了，早上五点多起来，从家开到了车管所，办完手续后又战战兢兢地开回了家。这是第一次单独一个人上路，没磕没碰没撞没出事儿。好吧，成功的经验会让你找到自信，于是后来我就敢上路了：怕啥，就那样嘛。

程序员的情况与此类似。比如你让一个 C++ 程序员去开发一个 Android App，他可能只需要两个星期就能学会 Java 和 Android 开发，做出一个 DEMO。然后呢，跨过门槛，一切障碍都不存在了，前面都是坦途，没什么能阻碍这个程序员解放全人类的伟大理想了。

这种体验的不断重复使我们由衷地相信好的开始是成功的一半，所以，当我们哪怕仅仅做出一个小小的 DEMO 程序，也会兴高采烈地发布我们的结论：啊，我们已经突破了关键技术，从已有的 DEMO 来看，技术上的问题不再是问题了。

当然事实并非如此。前路险滩无数，泥沼遍布，暗坑到处都是，当你迈出第一步，欣喜地以为天下独握我手一切尽在掌握的时候，那些你难以预料的艰难险阻已经出发，快速向你围拢过来。

我开车还没超过 100 公里，就遭遇了第一次交通事故，和保险公司打了回交道。作为程序员，你说难题已被攻破，技术上已经没有风险，其实仅仅是你过于乐观了，真正的旅途还未开始，让你挠头的事儿还未到来。

当然，程序员是乐观的，他们说，无论前路如何，只要逢山开道遇水搭桥见佛杀佛见鬼杀鬼即可通关，天塌了有个儿高的顶着呢，项目黄了有项目经理扛着呢，公司关门了大不了换个地方再战江湖，我有手艺，到哪儿混不来一碗饭吃？

好吧，我们就是如此乐观。如果你要和程序员打交道，一定要谨记此点，他们的职业习惯使得他们倾向于把事情简单化，只看好的一面，未知的风险他们会一笑而过。所以，你要有预期，他们常常前半程轻轻松松高歌猛进，后半程心情沉重步履维艰。

## 怎么告别“混日子”

( 写于 2015-03-18 )

我在“咦，你也在混日子啊”一节中描述了一种混日子的现状，有朋友说其中只提到了设置目标告别混日子而没有展开论述“如何设立目标”“怎样为目标而努力”这两个关键问题，也就是说，我提出了问题但答案没有落到实处。这次呢，我决定认真地来谈一谈如何建立目标，如何为目标而努力。一切从个人经验出发，仅供参考。

### 目标是什么

“目标是个人、部门或整个组织所期望实现的成果。”

这个定义看起来冷冰冰的，那还有另外两个更感性的词语：梦想、理想。但这两个词语似乎又过于虚无缥缈，谈起来感觉就像遥遥无期永远无法实现的东西。

有理想才有目标。

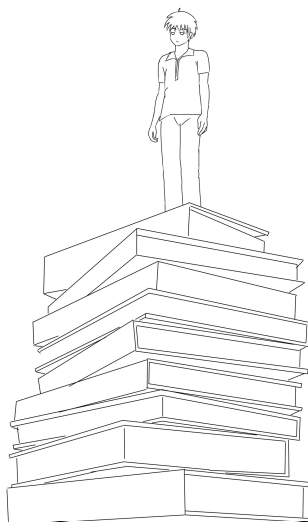
理想和目标是密不可分的，结合在一起讨论会更有意义。

**有理想才有目标，有目标才有行动，有行动才有结果。**

这句简单的话，我以后还会提到，它阐述了理想、目标、行动、结果这四者之间的关系。

你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 如何寻找目标



如何寻找目标？这是非常关键的一点。

感觉越写越正式，这偏离了我的初衷，有点惴惴不安，也有点不适应。

我把程序员的目标分为三类：公司目标、职业目标、生活目标。

王江民开辟的“白衣飘飘行走江湖”的时代已经远去，一般的程序员很少能够单枪匹马干出一番事业，多数时候我们只有把自己融入到某个集体中，借助公司这个平台，才能实现自己的职业目标和生活目标。因此，我们先来看看公司目标。

### 公司目标

程序员是公司的一份子，参与到实现公司目标的过程中。这个没什么好说的，食君之禄忠君之事，拿了人家的钱，就是要给人家干活儿的。

美国管理学家彼得·德鲁克在其 1954 年出版的《管理的实践》一书中，首先提出了“目标管

理”的概念，然后目标管理（Management by Objectives, MBO）大行其道，到现在依然被广泛使用，有非常多的公司在使用 MBO。我之前所在的西安信利软件科技有限公司，也曾经推行过目标管理。当时还学习了另外一本书，串田武则（日）的《目标管理实务手册》，感觉大有收获。

目标管理有 5 个要素：

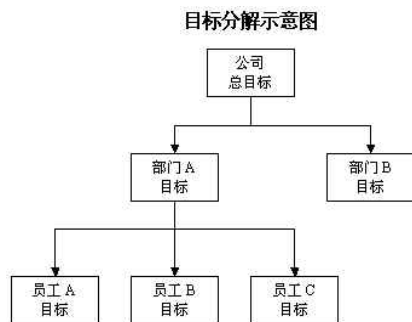
1. 什么目标
2. 达到何种程度
3. 如何实现
4. 何时完成
5. 完成得怎么样

通常公司在实施 MBO 时，还会和绩效挂钩，目标完成得好，绩效结果就好，奖金就多，升职加薪就快。

公司目标，一部分来源于远景规划、环境变化，这是纵向的；还有一部分则是横向的，来源于部门之间，不同部门有交叉协作，A 部门的目标要达成可能需要 B 部门协助，或者多个部门在一条产品线上，它们的目标有依赖关系。不管是纵向的还是横向的，都需要自顶向下分解，大目标分成小目标，最后落实到具体的员工来执行，程序员就是负责执行的员工。

公司目标最终分解到个人，产生了个人的工作目标。

这里是一张示意图：



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

如何寻找公司目标、如何分解、如何制定措施、如何执行，我是半吊子，具体就请参考《目标管理实务手册》吧。这里要提的是：

程序员作为公司的一员，必然要参与到公司目标的实现中来，公司成功，个人才可能成功，所以一定要搞明白自己对公司目标能有什么贡献，积极地去实现由公司目标而来的个人目标。

假如你在一个公司，都不知道要干什么，那要么是公司管理出了问题，要么是你的领导不合格，这两种可能性会比较大。如果是这种情况，那你多半可以考虑离职了——在一个不知道要干什么的公司，很难干出成绩来。

## 职业目标

职业目标是指人在选定的职业领域内所要达到的具体目标。

这是每个人都要想的问题。

我在面试时经常会问应聘者未来三到五年的规划，这个问题实际上针对求职者职业目标中的中期目标。有的程序员会告诉我说，三年后他希望成为项目经理；另外一些程序员说他希望五年后成为视频解码方面的技术专家；还有一些程序员会不好意思地笑笑，说还没想过这个问题。你是哪种呢？

职业目标分为短期、中期、长期。

短期目标一般是 2~3 年以内的目标，比如你刚进入移动开发领域，那你可能希望花 2~3 年的时间熟练掌握 Java 语言、Android 应用框架的各个方面以及系统框架的某些部分，能够独立完成较为复杂的 App 开发。

中期目标一般是 3~5 年内的目标。

长期目标，5~10 年的规划，一般设定较为长远的目标，比如你希望 30 岁时可以成为 IBM 西安分公司的一线经理，或者希望成为某个小型公司的技术副总。

还有更为长远的规划，那就是：人生规划。

说来说去，忽然发现我自己就是一个职业目标不明确的人啊，冷汗一身。

好吧，我开始为自己辩解了，希望你不要这样。其实，我们真能看那么远吗？

现在有一种职业，叫作“职业生涯规划师”，这个还可以考 **GCDF** 证书。

假如你有自己的职业目标，那当遇见在一个公司没事儿可干的情况时，你就能非常清楚地意识到自己应该离开，果断地离开。如果碰到间歇性空闲，那职业目标也会帮助你在空档期建立一些有助于实现职业目标的短期计划，让你不至于拔剑四顾心茫然，浑身没个着力处。

## 生活目标

你看那么多小说、散文、身心灵书籍，心理学书籍，励志书籍，都企图说明白这个问题，可是，从你的角度看，谁说明白了？所以我也不大可能把这个东西说明白，我只有一句话，还是舶来的：

我们在生活中所做的各种努力，多数是为了让自己有归属感和价值感。

我其实是一个懵懂的人，当我决定做一个程序员时，像“我就适合干这个”“我就喜欢干这个”“我能干这个”之类的感觉和判断都没有，我的出发点相当朴实：在上一份工作中我感觉不到自己的价值，干得没劲。

所以，从个人经验出发，就工作而言，我觉得有时不是你觉得新的工作特别适合你、能让你一展所长，而是你对上一份工作的现状感到强烈的失望导致你愿意花一些机会成本来尝试一下。如果你新公司所做之事与你个人的职业目标方向契合度高，那就更好了。

工作即生活，工作和生活不是割裂的，应当是和谐统一的。

工作是生活的一个重要组成部分，占据了我們年富力强的那段岁月的大部分时间。当我们考虑生活目标的时候，很可能就包括了工作目标。比如你想成为一个唱作俱佳的歌手，那唱歌和创作就会成为你的工作也会成为你的生活。比如你想成为一个伟大的程序员，写出一个新的移动操作系统，把 **Android** 和 **iOS** 拍死在沙滩上，那估计你的生活也会因此发生翻天覆地的变化。



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

我们来说普通人吧，像我们这种程序员，大部分都是一般人儿，生活平淡、平凡、平常、平实，那么，我们的生活目标是什么呢？

“年轻时拼了命赚钱，年老时拼了钱保命”，这当然不是我们推崇的生活。

“起得比鸡早，睡得比狗晚，干得比驴多，吃得比猪差”，自然也不是我们想要的。

“买房，买车，买女（男）朋友，养小宝宝……”，是这个吗？

.....



在我看来，这都是说不清道不明的东西，你喜欢干这件事儿，他愿意干那件事儿，你钟意一掷千金花天酒地的生活方式，他甘于一簞食一瓢饮的日子，都挺好的。如果你能做到“采菊东篱下，悠然见南山”，我会非常羡慕，甚至还会有嫉妒。

时间永是流逝，而未来不会自动到来。未来需要我们自己去创造。简单点说，我们就是通过实现一个目标又一个目标来抵达并创造未来的。没有目标、规划，就没有你想要的未来。

有时我们看不了太远，那就从近期的目标着手，看看自己今年、明年想做的事情都有什么，在一张纸上列一列，分析一下。哪些事儿不做就会让自己后悔，那这些事儿就具有重要的意义，要制定计划去完成它们。哪些事情做了会让我们更有成就感，那也可以排在前面去实现。哪些事情具有递进特性，可以延伸出新的目标……分析之后，你就可以确定那么几件必须完成的，这就是你的目标了。

**生活目标其实就是一件如果你愿意投入精力去做，就可能达到的事。**

比如你想今年考个驾照，明年买辆车，也算一个非常具体的目标。

比如你希望用三年时间完成一次职位晋升，从程序员变成软件项目经理。

今年再写一本技术周边的书，这是我的一个目标。

只要你愿意正视自己并花时间去分析，一定会找出一些你想做的事情。而这些事情，能让你有归属感和价值感，就可以成为一个好目标。

最后，我想说的是，不论你追求什么样的生活方式，你的生活目标是什么，最终落脚点，其实是“心”，当你或舒缓或紧张地随着时间的河流抛洒自己的生命时，只要你感到心安，感到喜悦，感到平和，这就足够了。

## 为目标而努力

目标就是计划。

有目标无计划，一定白瞎。

要实现目标，得坚持、努力执行你的计划。

## 工作目标

前面我们谈公司目标时提到公司目标会分解到个人，还提到 MBO。MBO 里其实不单有

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

目标，还有实现目标的措施，以及对这些措施的管理。如果你是个普通的程序员，公司或者你的上级应该辅助你制定这些东西，如果你们公司不用目标管理法，那我建议你自己搞明白目标管理法是怎么回事儿，因为它不仅适用于工作，也适用于生活。

我们以开发一个推送服务为例来说明如何建立目标和计划。

目标是：开发一个推送服务。这个有点笼统，还需要把服务指标细化一下：

- 支持 Android、iOS 两个平台
- 支持亿级用户
- 消息可以在 2 秒内抵达手机
- 消息抵达率为 99%
- 服务要 7x24 小时运行，故障恢复时间小于 1 分钟

现在服务指标有了，假如你是项目经理，你的 MBO 卡中“开发推送服务”这条内容基本就 OK 了，再填入实现措施、预期时间什么的，就完整了。

而要弄明白实现措施和时间，我们还需要进一步来考虑实现措施。

在制定实现措施前，这些需求还要进一步细化。我们以第一条，支持 Android、iOS 两个平台为例来说明。

推送服务应该分为服务端和客户端。

服务端是不需要跨平台的，你可以使用 Windows、Ubuntu Server、Cent OS 之类。服务端还需要管理界面，能让运营人员从管理界面上编辑并提交推送消息。估计还得要 API 接口，可能有些程序会自动产生要推送的消息。

服务端和客户端需要通信，你要定义通信协议。

Android 需要实现接入模块，必须是 Service，还得应对待机问题、第三方安全软件查杀问题。iOS 必须考虑到苹果的限制，可能要与 iOS 内置的 push 机制结合。

这里面事儿挺多的。再挑一个来细化，拿通信协议说吧。

通信协议有开源的，比如 XMPP，MQTT，当然也可以自己定义。那选择哪个呢？你要

开放给第三方使用吗？

不管你顺着哪个需求点梳理下去，都会分解出很多功能模块，每个功能模块又可以再拆分。在工作上，你要想实现一个目标，必然得经历目标分解的过程，把大目标分解到一个又一个小的、可以在两三天就能实现的任务，这样才具有可执行性。然后把这些任务排排座，每个任务都给个时间盒，计划就出来了。最后就可以按计划来执行你的项目、完成你的目标了。

## 如何有效利用空档期

本节起源于“如何告别混日子”这个问题，那么，问题来了：

在工作中，没事儿干的时候该做什么？

前面谈职业目标时提到，清晰的职业目标会帮助你度过工作期间的闲暇时间（如果有的话）。

举个我个人的例子吧。

2014 年年中，我负责的产品终止开发（只剩维护），有几个月的时间没什么具体的事情可做。晃荡一天可以，晃荡时间长了这心里发慌啊。幸好我对技术情有独钟，也比较爱钻研，于是，我就想看看之前用过的框架——Qt——怎么样了，于是就去研究，发现它居然支持 Android 了，也支持 iOS 了。兴奋的同时，我立马下载最新版本，搭建环境，开始写代码做实验，总结谁，分享到博客上。

就这么着，继续学习技术，帮助我度过了原本可能无所事事的几个月，庆幸时光没有白费。那段时间的研究，还机缘巧合出了本书：《Qt on Android 核心编程》。说实话我很享受那个学习、写书的过程。天天晚上熬夜写，天天早上早起写，周末窝在家里调试代码，乐而不觉其苦。很多事情都是这样子的，你喜欢干就不觉得苦。当我写完书时，反倒觉得空荡荡的。

其实我们在开发过程中，会用到各种各样的技术，多数时候为了赶项目，都是以实现功能为第一要务，只见树木不见森林，用过了就扔一边了。新项目来了，急匆匆地又忙活去了。但你要想提高，就得回顾、总结、拓展，把用过的那些技术一一列出来，在闲暇的时候，以

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

你用过的那个技术点为中心来画同心圆，一层层向外扩展，丰富你的积累。这是不辜负大好时光的好办法。

也有程序员的中期职业目标是做技术管理，那如果你有空闲的时候（没空闲也要挤时间），就可以看看项目管理方面的书籍，比如《项目管理知识体系指南（PMBOK 指南）》《项目管理：计划、进度和控制的系统方法》《人月神话》《带人的技术：不懂带人你就自己做到死》《软件工程（实践者的研究方法）》《项目管理修炼之道》……让自己对项目管理和团队管理有个理论上的认识。相关书籍很多，简直可以用汗牛充栋来形容，我列出的那些不过九牛一毛，更多的书籍你可以利用亚马逊之类的网站的相关推荐来了解。

## 新目标的诞生

生活就像翻山越岭，一山过去还有一山，我们就这么一直不断地攀登。在攀登的过程中，有时你会发现新的山峰，新目标就诞生了。

在我写作《Qt on Android 核心编程》的过程中，我发现 Qt Quick 这个框架简洁、优雅、强大，而我没有照顾到它，心里总不是那么安宁，觉得这么好的东西应该给它一点关注。于是我就定了目标：研究 Qt Quick，写一本专讲 Qt Quick 的书。

有了目标，我很快从写完《Qt on Android 核心编程》的空虚中走出来，把自己的闲暇时间都投入到研究 Qt Quick 中了。

选定要讲的内容，列出大纲，制定计划，评估每个主题预期花多长时间，写成文章又花多长时间，最终制定了一个粗略的计划，准备在三到四个月内完成整个工作。

有了计划后，我就梳理安排自己的时间，把晚上、清晨、周末都用上，一个主题一个主题地去研究，学习时记录自己的心得并保留示例程序……

3 个月过去后，我积攒了大量的原始资料，《Qt Quick 核心编程》也基本完稿了。

没人要求我写书，这都是因为我本身喜欢写作又喜欢技术，是一种自发的行为。研究、学习技术的过程让我感到舒服，写作让我觉得自己有价值，所以我愿意做这样的事情，哪怕

金钱上的回报寥寥也不以为意。

## 行动起来

最后的最后，我们必须强调的是：一切目标、计划，必须落实到行动上来，否则都是空谈。个体之间为什么有那么大差异？除却“拼爹”的因素，最重要的一点就是：有的人坐而论道，有的人用行动证明自己。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 一个老程序员的 2014

人生天地之间，若白驹过隙，忽然而已。蓦然回首，头上似霜雪，脸上似山川。我的 2014 就这么毫无眷恋地离我而去了。既然往事已不可追，未来尚未到来，在这年末岁首的间隙，就让我来回忆一下余烟尚存的 2014 吧。

2014 年，对我而言，发生了很多事，不管怎样，我还是一个程序员，这是工作、生活，以及我自己，给我的标签。也许很快这个标签会泛黄、蝶化、零落成泥碾作尘，不过不论怎样，它是我青春的纪念，它仍将伴随着我继续前行。

### 坚持技术博客写作

看到这里，也许你笑了。

是啊，这一年我不务正业，虽然每周都会更新博客，但是技术含量越来越低了。

2014 年，我在 CSDN 开了好几个专栏，没人邀请我，是我死乞白赖找 CSDN 申请的。有这么几个：

- C 语言面向对象编程
- Qt on Android
- Qt Quick 简明教程
- 漫谈程序员

我在这上面花费了大量的时间和精力。有个面试官曾问我，你怎么平衡写博客和工作？其实你看我的很多文章，要么是大清早，要么是晚上，要么是深更半夜。所以，我以为，这是不会影响工作的。

以前我干事情总是三天打鱼两天晒网，随性而为，不想弄了就停了，今年我希望有些东西可以坚持下来，持续地去做。于是我就这么坚持下来了。关于写博客有什么好处，真是一言难尽，您看一看“技术人员为什么要写博客”这篇文章就知道了。

在这一年里，我在 CSDN 的技术博客排名，由于我出乎意料的勤奋，从万名之外上升到了千名之内。

至于明年还会不会这么写下去，那就两说了，不过“漫谈程序员”系列，至少还会更新 3 个月。

## 独立写了两本书

简单介绍一下我的书：

《Qt on Android 核心编程》主要分两部分，一部分讲述 Qt 核心的概念，比如信号与槽、元对象系统、网络、Qt Widgets 等等，非常适合 Qt 初学和进阶；另一部分讲解 Qt 在 Android 上的开发主题，比如触摸、手势、传感器、JNI 扩展等适合对使用 Qt 开发 Android 应用的感兴趣的朋友。总之这本书既可以作为 Qt 入门参考，也可以作为 Qt 移动开发的入门参考。

《Qt Quick 核心编程》是专门讲述 Qt Quick 技术的，从 QML 语言、Qt Quick 基本元素、Model-View、动画、多媒体、网络、Android 主题……该讲的都讲了，不该讲的也讲了，总之我觉得它是业内第一本系统讲述 Qt Quick 的书，是填补空白之作，对于初学 Qt Quick (QML) 的开发人员和想进阶的开发者的参考、借鉴意义。

CSDN 因为我的书，对我做了一次专访，题目是“专访安晓辉”。

有一天我爱人问我：你写这两本书花了多长时间？

我掐指一算，从开始写到交稿到修改到出版，前前后后七八个月。

我爱人说了一句话：没想到我忍了你这么长时间了。

瞬间“石化”。然后又融化。

是啊，我给妻子、孩子带来多大的影响啊，我夜以继日地写，牺牲了陪女儿的时间，牺牲了陪老婆的时间，牺牲了娱乐的时间，图什么呢？我爱人说，难道就为了你的一个梦，就牺牲了我们这许多人？是啊，我太自私了点儿。所以，我决定今年不再写书了。



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 录制了一门视频课程

好吧，我起早贪黑，花费了很多精力录制了一门视频课程——《Qt 基础与 Qt on Android 入门》。这是与我的书相关联的，但又是独立的一件事情，也是我从未做过的。

很多个早上五点多，很多个晚上十二点，我坚持下来了。

## 经历四家公司

我一直以为自己是一个很稳定的人，不幸的是，我今年让很多人失望了。

今年我历经了四家公司，变换了  $N$  种身份：部门经理、总监、高级开发工程师、技术图书作者、CSDN 上坚持扯淡的程序员、讲师……惊讶吗？

现在呢，我是一个初创公司的技术合伙人，一切从零开始，我们要来一次说走就走的旅行，而结果，是未知的。未知，才美丽。

2014 是我工作、生活变动最大的一年，再不折腾就老了，这是我这一年的注脚。

## 尝试几种新的平台和语言

作为程序员，在技术方面，2014 年的工作真有点儿乏善可陈了。

我仅仅是在一些方面做了一些尝试（都没有深入）：

- 学习了 Qt Quick
- 使用 Qt 在 Mac OS X 上开发
- 使用 Qt 在 iOS 上开发
- 学习了几天 Objective-C 和 iOS 开发
- 学习了 JavaScript
- 学习了几天 Node.js
- 试用了一下 Express
- 学习了几天 Scala

- 试用了一下 Play Framework
- 听人讲了一次 Docker

## 开发圈子

一个人的战斗是孤独的，程序员也是社会化的动物，我们需要与他人同在，我们需要帮助他人，也需要他人的帮助。

我申请了 CSDN Qt 版的版主，我加入了多个 Qt 开发群，我在博客上分享我的技术心得，我还主动将线上的关系延伸到线下。而在以前，我不会去做这样的事情，现在，我觉得这是很自然的事儿。

## 拿了驾照

好吧，不解释。

## 出国旅游了一次

虽然去的是泰国，毕竟还是出国了，对于我这个“土包子”，也是一次难得的经历。

## 终于看到了女神

林青霞来西安签售，排了一下午队，终于看到了女神，买了本《云去云来》，有签名。

平生第一次，特此记录。

## 看过的几本书

书越看越杂了，有的看完了，有的翻了翻。

- 《目标管理实务手册》
- 《首先，放下一切常规》

🗨️ 你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

- 《高效能程序员的修炼》
- 《高效程序员的 45 个习惯》
- 《JavaScript 高级程序设计》( 未看完 )
- 《Node.js 开发指南》( 未看完 )
- 《Programming in Scala》( 未看完 )
- 《iOS 编程》( 未看完 )

.....

列出来才知道，原来还是看了一些书的，总算可以坦然面对“时间都去哪儿了”这个问题了。

## 最重要的变化



上面我列了很多事情，还有很多未能罗列的，一年之中事情太多太多，多到我已经忘记了 90%的时间是如何度过的。

我们每个人都有个记忆的筛子，筛去那些不重要的，留下那些在我们心上刻下痕迹的。

2014 年，对我来讲最重要的不是那些具体的事情，而是：

**我终于想通了生活与工作上的一些事。**

在我 34 岁的时候可以想通工作、生活、梦想这些事情，虽然有些晚，但我依然感到高兴。

想通的结果是，做事不再过分执着于形式，他人的看法也不能再对我产生致命的影响，虽然我还不能像诗里说的那样“世路如今已惯，此行到处悠然”，但也可以慢慢地跟随心的方向了。

也许旁人看起来我在折腾（我爱人有时也这么看），昨天很执着地要告别这个，投奔那个，过两天又要告别刚刚投奔的这个……不过对我的内心来说，这都是必然，我已经有了一张可以指引我的心灵地图，我每一次的决定，都是坚决的，是受到了发自内心的某种指引，而不是人云亦云。我能够用属于我自己的经验与认知来解释这不断发生的一切，并且，不悖于心。

以后还会发生很多事，或许我还会有很多的犹疑与彷徨，但我相信，我已经有了坦然面对的能量。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 坦承无知有多难

承认自己的无知到底有多难？

### 我被问过的问题

“总觉得有义务给每个问题找到答案，哪怕不懂，也要装懂。”

这是不少程序员的想法和做法，出于自尊心也罢，担心被别人窥见自己的无知也罢，担心别人认为我们藏私不愿分享也罢，反正我们经常这么做。

回忆一下我曾经被问过的问题吧。

（1）什么是大数据？

大数据就是数据量很大。

（2）什么是云计算？

云计算就是许多服务器协同运算。

（3）什么是并发编程？

用多线程或者多进程，让任务并发执行。

（4）项目计划有什么用？

保障项目按时完成。

（5）什么是项目管理？

就是带项目，管好项目。

（6）好团队的定义是什么？

就是战无不胜攻无不克的那种团队。

( 7 ) 什么是跨平台开发?

针对多个目标平台进行开发。

( 8 ) select 模型，单机，能处理多少并发连接?

3000 吧。

( 9 ) Qt 比 MFC 好在哪里?

跨平台，面向对象。

( 10 ) Android 为什么选择 Java 作为应用开发语言?

Java 学习门槛低。

( 11 ) 怎么评估一个程序员的绩效?

360 度考评，或者 KPI，或者 MBO，或者。

( 12 ) 什么是敏捷?

SCRUM 或者 XP。

( 13 ) 怎样做工作量估算?

项目经理和高级程序员给大家估算。

( 14 ) 阿猿表现怎么样?

还不错吧。

( 15 ) 还有多久 Bug 才能收敛?

两个星期吧。

( 16 ) H.265 比 H.264 好在哪儿?

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

编码效率更高。

(17) 你是干什么的?

软件开发。

软件开发是干什么的?

开发软件的。

.....

我被各种各样的人问过各种各样的问题，各种各样的答案都给过，上面仅仅是沧海一粟，答案也千奇百怪，多数时候都不准确，想想真让人难堪。

可是我很少对别人说：我不知道。

承认自己无知需要勇气。

“行，我知道了。”

“我知道。”

“我懂!”

“谁说我不懂?!”

.....

我们日常生活中与人交流时，经常会用到上面的话，想想是不是?

人人都愿意自己成为一个“知道很多”的人，很少有人乐意成为一个“愚昧”的人，因而我们经常让自己扮演“全知神”，不断地抛给别人模棱两可的答案，不断地用看似正确的废话来掩盖自己的无知。

我觉得这里面比较直接的一个原因是：面子。

## 无知是一种客观



一个人到底是有知还是无知，由谁说了算？

我承认“不知道”，就是“无知”吗？我说“知道”，给出一个解释，就是“有知”吗？

这其实都不是问题。

其实，“吾生也有涯，而知也无涯”，一个人穷尽一生去学习，也还是有很多不知道的东西，更甚之，知道的越多，会越感到无知。

苏格拉底更精辟了：“我唯一知道的事，就是我一无所知”。

说到底，无知是一种必然，对此有个客观认识就是了。



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 坦承自己的无知

生命有限，时间有限，我们被有限的枷锁套牢了，注定了这辈子摆脱不了“无知”。

承认自己的无知是对自己的一种诚实，一种肯定，一种警醒，也是一种鞭策。

每个人都是从无知开始的，知识并不会自动进入到我们的脑袋，我们要付出心血、勤勉努力才能牢牢掌握知识，而这一切，都始于承认自己的无知。从这个角度上讲，“无知即力量”，只有我们承认并且勇敢地面对无知，才能鞭策自己，将无知转化为前进的动力，真正弄明白自己不知道的事情。

懂得愈多，就愈能体会其实自己知道的事物宛如沧海一粟。懂得愈少，就愈会产生自己什麼都知道的错觉。所以，当别人问了我们不懂的问题时，抛弃饰过的惯性，马上承认“我不清楚”、“我不知道”，然后，想办法搞懂它吧。

就这几点，提高工作效率🐼

## 就这几点，提高工作效率

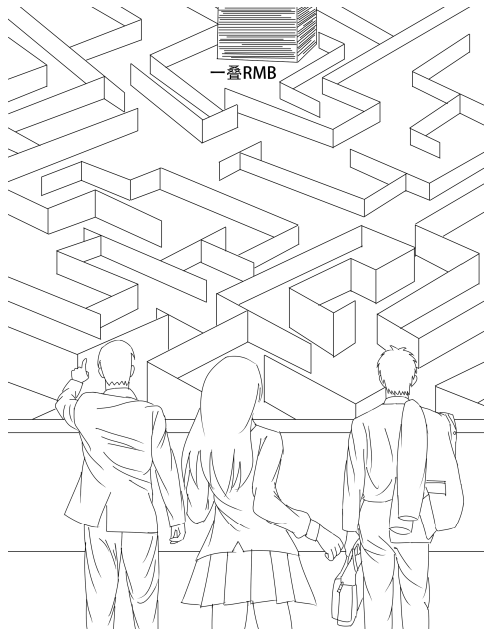
有一本书，《高效能人士的七个习惯》，建议希望提高工作效率的程序员阅读一下，必然会受益良多。

要想提高工作效率，不论你看什么书，看什么文章，只有下面这三点最重要：

- 改善的决心
- 剖析自己，找到改善的切入点
- 付诸行动并且坚持

现在，让我们步入正题吧。

### 目标驱动



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

有目标才能高效，漫无目的的布朗运动是消耗生命的最佳手段。

我们爬山，山顶就是目标，心怀山顶，在登顶之前，总是充满希望，纵有沟沟坎坎，也会克服困难，努力前行。

作为一个有追求的程序员，心中应该不只有一座山，更应该有一条绵延不断的山脉，千峰错落，峰峦叠起，终成峻岭。设立目标，付诸行动，不断攀登，是自我成就的不二法门。

扯淡惯了老刹不住车，放到日常工作中，就这么三点，就能让你与众不同：

1. 每周一问自己：这周干什么，定下这周的目标。
2. 每天早上问自己一个问题：今天干什么，定下今天的目标。
3. 每天下班前花十分钟：今天完成了什么、有什么问题，记录下来。

其实这是我从敏捷开发每日站会的三个问题延伸出来的。我的观点是，敏捷不只是做某一个项目时的模式和手段，而应当是一个有追求的程序员贯穿始终的工作习惯。

## 谈一谈理想

其实这个应该放到第一条，所谓有理想才有目标，有目标才有行动，有行动才有结果。

唯有当一个人确实了解他自己想要过的是什么生活、他所要追求的目标到底是什么时，他才会觉得生命充实且有意义。

工作随着志向走，成功随着工作来，这是一定之理。

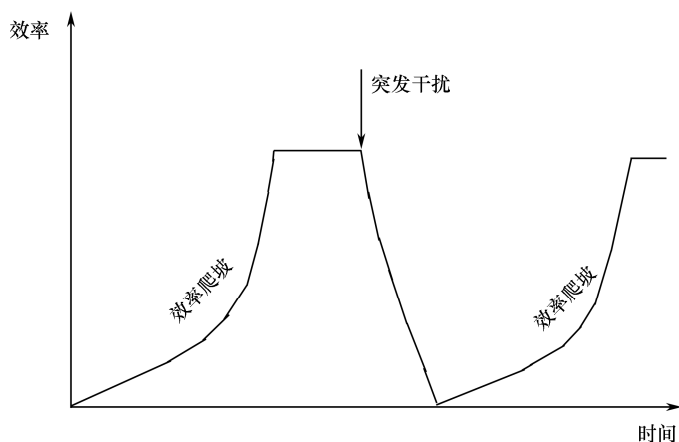
立志是事业的大门，工作是登堂入室的旅程。这旅程的尽头就有个成功在等着庆祝你的努力结果。

所以，我们还是谈一谈理想，找一找那个你放弃了会懊悔终身的目标。只有找到了它，我们才不致于在生活之路上迷失。只有找到了它，我们才能明了什么样的工作是我们需要的什么样的工作对我们整个人生有价值，然后我们才能坚忍地去追求，一步一步地前行。

## 从现在开始改变

作为软件开发工程师，在工作时需要高度集中注意力，动员所有神经来解决问题，最怕受到干扰。我正写代码写得 HIGH 呢，项目经理蹦过来问我进度怎么样，质管发邮件过来说过程检查不合格，测试 MM 说某个需求理解不了，有个无聊的网友在 QQ 发消息撩拨你晚上去幻影凶间，产品经理蹭到身边说不好意思兄弟需求又要改了……要知道保持速度与节奏很重要啊，我正突击呢你来这么一下子，看起来是几秒钟的事儿，可对程序员来讲，也许半天功夫就白费了，要想再找回断掉的思路，花费的时间可是那几秒钟的几十倍几百倍。

看看下面这张图：



“树欲静而风不止”。这是我们在开发过程中必须要面对的现实，各种各样的干扰会让我们的工作效率大打折扣。如果你有同感，往下看吧，保证不让你失望。

## 邮件的打扰

领导会要求你关注邮件，每隔多少分钟检查一下邮箱，最好安装一个邮箱客户端，设置一个自动检查新邮件。我原来也要求过部门的同事，现在想想这不大靠谱。关键的一点是：真正重要的事儿，真正紧急的事儿，并不通过邮件告知。所以，尽管关注邮件是必要的，还

💬 你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

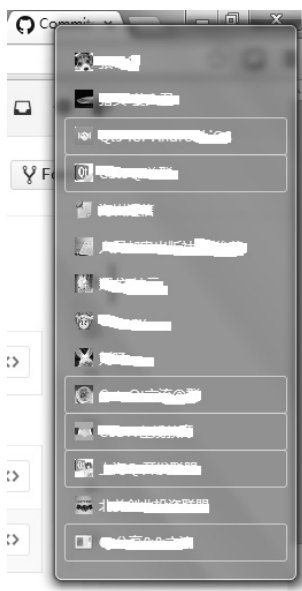
是要给它设置一个限度。照下面这么做就好了：

- 每天早上扫一下邮件标题，把重要的、紧急的标记出来。
- 每天中午查看邮件，汇总重要的事儿。

## 即时通信工具的打扰

这年头，人都忙着在网上社交了，很多精力和时间都贡献给网络上的虚拟圈子和对象了，其实，我要说的是，玩的是热闹也是寂寞，别上瘾为好。

看这张图片：



这是我某个时刻的 QQ 通知截图，同时和 14 个人聊天，有的人比我还多，同时聊三五十个，要是笔记本没有静音，“滴滴”之声就像农村河畔夜晚的蛙鸣，此起彼伏，连绵不绝，真够让人忙活的，你要是每听到一声“滴滴”都看一下，那完了，就算你开着 Qt Creator /Eclipse/VS/Android Studio 之类的 IDE，也没时间切换过去敲代码了。这场景，基本上就变成

就这几点，提高工作效率🐼

聊天为主开发为辅，淘宝小二的工作更适合你。

电脑上有 QQ，手机上有手 Q、微信、陌陌等，也是一会嘀一下、震一下，也许我们每天要看手机 100 次以上，重度手机依赖者在 300 次以上，看微信、看微博、看各种通知、看短信、接电话、看时间、看天气、看养的宠物、把玩游戏、看小说、看漫画……智能手机太富有侵略性了。假如你从口袋里掏出手机的次数远远大于别人给你打电话的次数，那你就存在过度关注的倾向，就需要你注意了，阿猿，你又没那么多女朋友，不会错过晚上的约会的，担心个什么劲儿呢。

下面几招就可以让你摆脱或降低即时通信工具和手机对工作的干扰：

- QQ，留给中午休息时间吧，或者下班前；关掉 QQ，或者关掉消息提醒、设置提醒周期为很长很长。
- 微信，关掉提示音吧。
- 微博，关掉提示吧。
- 手机，调整为震动或者静音。
- 陌陌，办公室都是熟人，关掉它吧。
- 智能手机上各种应用的通知，都使用安全软件将它们禁止吧，或者设置通知的提醒模式。
- 断网，给手机断网，3G/4G/Wi-Fi 统统不要打开。

## 张弛有度

工作一阵子（譬如 45 分钟、30 分钟）就休息一下，有张有弛，活动下筋骨，站起来走走，活动脖子、腰、手，转转眼球看看远方。总之，一个人能集中注意力的时间长度是有限的，你要说你连续高效工作四个小时，那我表示佩服。

为了能够张弛有度，程序员还需要做一些工作任务上的调整，把编码、设计、杀虫等工作的粒度细分，最好一个小任务能在几十分钟内完成，这样节奏就配合好了。

## 关于代码

程序员要写代码，这里有一些小贴士可供参考，也许能提高开发效率：

- 自测，每完成一个小功能都自测一下，看看这个功能自身是否正常，集成一下，看看有没有影响其他模块。开发人员的自测非常重要，编码期间频繁地测试、集成，会大大节省送测后回归的时间。
- 一个小功能完成后，提交到 svn 或者 github 之类的代码托管平台上，免得宕机、停电、误修改造成重复工作，对心情产生恶劣影响，这种影响甚至比外来干扰还要大，还要持久。
- 多与产品经理、需求人员沟通，完成一个功能就给他们看看，一方面是看这是否是他们期待的，以便及时调整，另一方面是让他们对进度有所了解，降低他们因未知产生的焦虑，同时也能降低他们干扰你的频率。
- 多与测试人员沟通，完成一个功能集就可以 build 一个版本给他们看看，玩玩儿，让测试人员熟悉产品，建立对产品的直观感受，也给他们时间写用例。
- 多与项目经理沟通，完成一部分功能，就可以给他们看看，让他们知道你在 keep moving，也许他们也有一些想法，可以就此交流，免得什么都弄完了有人对结果不满意，要推倒重来，浪费时间。
- 多与团队成员沟通，很多功能都是要相互协作的，让他们知道你前进到哪里了，和与你工作衔接的同事约定联调时间，相互发布进度。如果不能在预期时间内联调，要提前告知。
- 多和 UI 及美工沟通，做出能看的东西就让他们看，看看是不是他们要的效果，免得后期返工，浪费工作量。

## 知易行难

如果你发现自己的工作效率有待提高，并且拟定了一系列的措施，那就执行吧，坚持执行。想再多，落不到行动上，都没有用。

执行了，还要想办法度量效果。比如你觉得 QQ 影响你的开发效率，那就关掉它，坚持

就这几点，提高工作效率🐘

一周，看看有什么变化。要想度量效果也很简单，找一个时间跟踪软件，看看你每天花费在 QQ 聊天上的时间，花在开发工作上的时间，记录下来。然后在关闭 QQ 的情况下，再记录你花在开发上的时间。对比一下，关于时间投入这个维度的结果就出来了。而工作成果这个维度，就看你自己如何度量了，比如通过工作量、完成的任务数等都可以。

总而言之，如果你觉得某个因素使你效率低下，那就制定策略，落实到行动上来，一定有效果。可惜的是有时我们明知那么做会有改善，可就是不够勤奋。



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 程序员的第二生活支点

早些年我在文学论坛写小说时，有朋友评论说我属于“存在主义”这个派别，总是追问生活的意义、存在的价值。我觉得这些问题是要想明白的，明白了才能更好地生活。下面咱就来说说程序员的工作与生活及其意义所在。

### 工作的双重意义

你为什么要来搞软件开发？

怎么老是问这个问题，好像问了好多遍了。

对啊，为什么当程序员？

我解释一下为什么我老这么问。这是因为，不论何时，我们都不应当忘记做一件事的初衷。因为随着你不断的前进，不可避免地要偏离你最初的航线，这样的偏离，有时是巨大的，甚至是与你的初衷背道而驰的。如果我们不停下来思索、审视、自省，我们就很难知道今夕何夕、路在何方。也许你成了自己当初讨厌的那个人却浑然不觉。

### 工作是赚钱的一种途径

1997、1998 年那会儿，程序员的薪资相当高，计算机大热，计算机软件大热，大专院校录取分数线，数计算机相关专业的高。这大部分是因为搞软件开发薪水高，吸引力巨大，我们的父母、师长、朋友有相当一部分都是看着“钱景”帮我们报志愿的。

现在呢？平均看，程序员依然是收入较高的。当然有各种吐槽，觉得程序员很“苦逼”，很“屌丝”。

对于既不是官二代又不是富二代的普通人来讲，工作的一大目的就是赚钱，进而养家糊口，最终让自己在物质上好过一些。

工作让我们在经济上独立，不再依赖父母、亲友，可以自己养活自己，这是一件多么了不起的事情，意义巨大。

只有经济独立了，其他事情才谈得上，女朋友、丈母娘、男朋友、房子、车子，都建立在经济基础之上。不是所有的东西都要有，不是所有的东西都和幸福相关，但路漫漫其修远，我们不能没有钱。所以，要做一个独立的人，在经济上必须要独立，必须要工作，这样腰杆儿才能硬起来，才能活出自己。否则，要么被别人鄙视，要么让父母在心里默默地叹息。

从这一点来看，我们为钱而工作，我们干软件开发为高薪，是自然而正常的。

## 工作对心灵的影响

我一向是个务实的程序员，不喜欢“虚”、“假”、“空”，喜欢脚踏实地干实事儿，可是这一次，我竟然要来谈谈精神领域的问题了。

问一个贼俗的问题：你快乐吗？

我不需要你的答案，只是想唤起你的思考。

还有一个问题：你幸福吗？

我是认真的，我的问题很严肃，咱们还是用正儿八经的语气来讨论这个问题：在工作中，你感到快乐吗？

如果你这么问自己，就会发现工作的另一重意义：

**工作应该带来满足、充实、快乐，有助于我们的心灵成长与心智成熟，让我们的生活更有意义。**

假如一份工作带给你的痛苦、纠结多于快乐，那无论它能给你多么可观的薪水，都不是一份好工作。在“自我发现，找到适合自己的职位”中，我已经提及如何选择让你感到平和、安宁、快乐、充实的工作，感兴趣的可以跟过去看看。

## 你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

作为一个程序员，我曾经被亲友问过这些问题：

- “考试成绩怎么样？”
- “找到工作了吗？”
- “一个月赚多少钱？”
- “是领导吗？”
- “有车吗？”
- “有女朋友吗？”
- “买房子了吗，多大？”
- “什么时候结婚？”
- “要不要二胎？”
- “孩子准备上哪个小学？”
- “孩子成绩好吗？”
- .....

有女朋友吗？

现在月薪很高吧，多少啊？

年终奖多少啊？

啥时候准备结婚啊？

在哪个名企工作啊？



可是很少人问：“**你过得开心吗？**”

世俗生活的价值观就是如此，我们身边的亲友，很少人关注你的心灵，更少人关注你过

得是否快乐。而我们自己，却不能忽略心灵的问题，一定要关注自己的内心。当午夜来临，当我们从白天的工作中隐退，当夜色笼罩了这个钢筋水泥的丛林，我们需要问自己一句：你快乐吗？

我们每天有超过 8 个小时在工作，工作除了带给我们经济独立，还给我们的心灵带来重要的影响。有的工作让我们感到快乐、充实、干了还想干，让我们容光焕发；有的工作则让我们痛苦、纠结、担忧、抗拒、忧郁，让我们忍不住想逃避，让我们每一个周末结束都感到茫然和犹疑。这种泾渭分明、截然不同的影响，正是我们判断一份工作是否适合自己的重要依据。

从这一点上来讲，工作即是生活的重要组成部分，一定要正视它，要选择让你感到开心（最起码不窝心）的那个。

## 第二个支点

虽然工作是我们生活的重要组成部分，但我们的生活里却不只有工作。

我不是说你还有家庭生活、有孩子、有老婆、有女朋友、有男朋友……这都不是我要说的重点。我想说的是：工作能够对我们的的心灵产生重要的影响，而影响我们心灵的，应该不只有工作，我们还应当找到其他滋养心灵的事情，至少一件，可以让你持续去做，持续助益于心灵的事情。

生活应该是多中心的，我们要欣赏多面的工作生活，并花时间发展自己的兴趣爱好。

这里的兴趣爱好，就是我要谈的，程序员在技术之外的第二支点。

## 平衡的必要性

科技是发展很快的玩意儿，程序员的工作节奏普遍较快，我们搭上了一列和谐号动车，而非绿皮火车。这样的高速列车上，哪怕你不情不愿，也会被裹挟着前行，你会落下很多东西。当你的速度、节奏很快时，惯性就会很大，而质量大的东西惯性大，对程序员来讲，肉身、开发任务都是看得见、质量大的东西，他们有强大的惯性，哪怕踉踉跄跄也会跟着列

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

车速度往前走。而心灵或者灵魂，是很轻、很慢的存在，它会在某个时刻被抛下，就像一片流云，慢慢地飞出窗外，而你的快乐、幸福，却更多地构筑在这些慢的东西上，所以呢，跑得快了，很多时候反倒感觉不到幸福了。

这个时候我们就需要平衡，我们要有快节奏之外的另一种生活，让我们的节奏慢下来，让我们能够有时间来关注心灵，或者修补快节奏工作对心灵带来的伤害。

我们工作挣钱，养活自己养活家人，有时难免会被这种生活的压力拉扯着做自己不愿意做的事情，难免不快乐。

如果一个人干的事情恰好是他喜欢的，又能带来可观的收益，养活自己、养活家人，那他真真是幸运儿。

而多数时候，我们没那么幸运。对于程序员来讲，这个问题更为突出吧。相当一部分程序员，是被各种压力、诱惑推动进入软件开发江湖来的，做着高强度的工作，过着快节奏的生活，经常是不满意的，也有人感到痛苦的；对那些并非由衷喜欢软件开发的朋友而言，这都是正常之极的反应，不必介怀，也没必要不好意思，为了生活，我们难免四处奔波和各种妥协：尽管工作带给我心灵一些伤害，可是它伴随的薪水于我却是必需的，所以尽管不那么开心，还是会干下去。

那么，在这样一种情况下，就要谈到工作与生活的平衡了。

## 寻找美好的事情

凡是美好的事情，都能给心灵带来裨益，让我们感到陶醉和满足。

有些人在我们眼里是工作狂，而实际上他享受当下的工作，工作对于他就是美好的事情，所以他会忘我，会发自内心地热爱。这也是幸福的事情，值得羡慕。

而对于不那么幸运的人，工作会对心灵带来伤害的人，则需要找到另外一件美好的事情，以便能够修复心灵的伤痕。

这就是我要说的第二支点了：美好的、能够滋养心灵的第二个生活支点。

俗话说不要把鸡蛋都放一个篮子里，对于我们自己这颗超级贵重的蛋，自然要更加重视，如果把鸡蛋放到工作这一个篮子里，万一工作不理想了，岂不是一切转瞬成灰？所以，我们要发掘自己的兴趣爱好，找到一两件自己爱干的事儿，让自己的心灵有其他寄身之处。

比如你喜欢旅游，比如你喜欢写作，比如你喜欢集邮，比如你喜欢当木匠自己做些小玩意儿，比如你喜欢航模，比如你喜欢阅读，比如你喜欢插花，比如你喜欢围棋，比如你喜欢做菜……太多了，只要你跟随自己的心，就能发现这样的事情。

我发现写东西是我的兴趣所在，也是我一直以来起早贪黑写“漫谈程序员”系列博文强大动力。你爱一事儿，就愿意去干，哪怕瀚海阑干百丈冰，也不以为苦，哪怕独上高楼望断天涯路，也不会后悔。这就是爱的力量，你我皆有，就看何时可以发现了。

“只要你有一件合理的事去做，你的生活就会显得特别美好。”

这是爱因斯坦的话，你我共勉。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 程序员的年龄怎么着了

我是一个老程序员了，我不断在思考一个问题：程序员的年龄天花板到底是多少岁？

### 其他行业里的老将

农村有一些老人 70 多岁了还能种地，不但能种小麦玉米，还能在自己房子后面种点白菜、萝卜、西红柿、豆角什么的，自给自足，不靠孩子赡养。

褚时健 70 多岁承包荒山开始种橙，现在“褚橙”是知名品牌。

我们家不远处有个同仁堂，同仁堂里有中医馆，中医馆里都是老先生，太年轻的大夫没人愿意挂他们的号。

### 老程序员啊

很多行业对年龄没有特别限制，有的行业甚至是越老越吃香，可程序员呢，很多人都觉得是吃青春饭的，英雄迟暮、年老色衰就会很悲哀。

果真如此？

### 国内环境

国内的 IT 公司，软件开发水平和意识整体较差，多数公司做出来的产品差强人意，一帮人做出一东西来，再一帮人花一大把钱去推广，就这么个现状。这种现状是由逐利的急切性引起的，逐利的急切性又客观上造成了对技术的短视，再往下推，自然就是对技术人员和技术能力的不重视，最后也就谈不上人才的培养和技术的积累了。

计算机软件已经成为整个社会发展的基础支撑，各行各业都需要，各行各业也都需要开发软件的程序员。行业背景不同，对技术的看法自然会有不同。我前面概述的现状，自然也有一些

公司不在其列，但 90% 以上是这样：东西做出来，钱赚到了，就是胜利，其他的，都不重要。

在这样一种心态和现实下，做东西的人，谈不上多受重视，很多公司的老板觉得拉一帮人就能把东西做出来，他认为要做的东西根本就没什么技术难度，网上抄一抄，加班搞一搞，这么着就出来了。年轻人身体好，牵绊少，能拼能加班，可以弥补与老程序员在水平上的些许差距，为什么不用年轻人呢？更何况年轻人又便宜，一个老程序员的薪水可能可以雇好几个年轻小伙子，三个臭皮匠能顶诸葛亮。

## 团队分类

为了讨论程序员的职业寿命，我们先得给公司或者团队分分类。

### (1) 外包型

什么是软件外包呢？

看看百度百科的定义：

“软件外包就是企业为了专注核心竞争力和降低软件项目成本，将软件项目中的全部或部分工作发包给提供外包服务的企业完成的软件需求活动。”

看见了吧，大多数时候，软件外包干的就是人家不愿意干的脏活、累活、低技术含量的活。我在“亲爱的‘程序猿’们怎么找工作”一节中已明确我的看法：有产品做，就不要做外包。

我见过的外包，有对欧美的，对日的，对华为的，这几类很多，前些年中国的劳动力成本低，外包业务很发达，现在印度成本更低并且软件开发规范程度更高，是比中国更受青睐的外包市场。

做软件外包的缺点有几个：

- 技术积累缺少持续性
- 不稳定
- 没有话语权

专门做外包的公司，有活儿了你就干，没活儿你可能就得滚蛋。所以这是不稳定的。



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

对程序员来讲，在外包类公司，是最容易遭遇年龄坎儿的。因为干的活儿技术含量低，个人的可替代性就强，经验丰富的老程序员薪资又高，每月 20K、30K 很常见，对老板来讲养不起，招几个年轻的，三千两千都有人干，还也能干得了。还有，你事儿多，有老婆孩子，孩子学校不定有什么事儿呢，晚上还得招呼孩子功课，加个班儿都不痛快。就算你技术很强，可外包这种性质，打一枪换个地方，活儿干完了就完事了，再强也没你用武之地呀。

## （2）项目型

中国电信、中国移动、中国联通，电信三家公司，养活了数不尽的项目型公司和团队啊。还有一众银行、金融企业、政企，也养活了数不尽的项目型公司啊。

什么是项目型公司呢？就是以给人做项目为主业的公司，接到项目就开工，有钱赚，比如你接人家电信的网上营业厅，比如你接人家某某大型国企的政务信息化等等；接不到项目呢，就收缩规模，裁员……我见过很多，电信、金融、石油、电力、教育、政务、铁路……各行各业都需要这种项目型公司，用自己公司的人，帮别人干项目，最终所有权不归你公司，归人家，可延续性很差，也不需要多少人来维护，就是要维护，留一两个核心的懂业务的人员就够了，其他人，鸟尽弓藏，没项目了你就只好走人吧。

项目型公司比外包型稍强一些，技术上可以自己选择，也能够某个行业长期积累经验，甚至你还有可能自己立山头自己干。不过对于不能拉队伍自己干的普通程序员来说，还是可持续性差一些，可替代性差一些。我就见过电信的项目，一个大项目拆成多个小项目，分给七八个小公司来做，今天你来做，下一期另外一个小公司和对口主管关系好，可能就把你替换了。有些程序员会在出没于电信大楼里的几个公司之间跳来跳去。从这一点上讲，程序员也不重要，公司也不重要，这个公司不干还有那个，这个程序员不干还有那个，另外程序员的价值也不高，有个懂业务的，配几个小年轻，事儿也就能搞成。

## （3）产品型

毋庸置疑，做产品的公司是程序员最好的选择。一个公司做了一个好产品，运营这个产品，产品会不断迭代、更新，程序员会跟着这个产品一直往前走，公司需要了解产品、了解业务、技术积累厚的人才，而要懂业务、懂产品、有积累，都需要时间，等你什么都了解差

不多了，也就是个老程序员了。而且随着你对产品了解得越深，你的不可替代性就越高，价值就越能凸显出来。当你的不可替代性更高时，老板就要考虑替代的成本。什么都不懂的年轻后生，根本撼不动你的地位。

做产品的公司有相当数量，也有很多知名的，比如腾讯、百度、阿里、爱奇艺、PPTV等，开发人员也数量庞大，几千到几万人的都有，但是这样的公司所拥有的开发人员总数量，与中国几百万程序员群体相比，还不是一个数量级的。那更多的程序员在哪里呢？就在项目型公司、外包型公司。

## 程序员的价值所在



对于程序员来讲，价值怎么体现？一个程序员的价值，就体现在他的稀缺性上。物以稀为贵么，亘古不变的道理。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

“闻道有先后，术业有专攻”这句话很适用于程序员，直白点说，技术的精湛不是以年龄来评判的，有的朋友在上大学时的技术水平已经比工作十来年的高了，有工作十年的程序员水平不如工作三年的，当然也有很多程序员的技术水平是正比于年龄增长的。

作为一个老程序员，只有你的技术能力正比于年龄，才是合格的，否则就必然越走越慢，慢慢掉队，最终就只能面临被年轻人挤兑、淘汰的窘况，结果就是被迫离开程序员的队伍。

程序员要想让自己的价值持续增长，有这么几个方向可供参考。

### （一）选行业

在一个行业内持续积累，对行业的理解到位，积累深厚，你的价值是巨大的。不信你去浏览招聘网站上的岗位要求，99%都要求相关行业背景。所以，选择一个靠谱的、前景好的行业非常重要，只要这个行业能够不断发展、前进，你的积累就是有价值的，你自己就是不断增值的。

### （二）技术

对于程序员而言，业务积累是屠龙刀，技术积累是倚天剑，如果两者都有，那就逢山开路、遇水搭桥，没有走不通的路了，这个程序员一定是先锋部队尖刀兵，到哪里都具有超强的攻击性。

技术上最起码的要求是技术水平和视野对得起年龄，如果有超越年龄的积累和犀利，那就没话说了，地球人已经阻挡不了你前进的步伐了。

就技术来讲，假如你掌握了一门很难又很重要的技术，那单单是这一点，就很有价值。比如你熟悉 Windows 内核或 Android 内核，能够熟练撰写各种驱动，那找个月薪五六十 K 的工作不成问题。再比如你图像处理算法很牛，再比如你视频解码算法很牛……像这些特定领域的算法，都是高难度高门槛的技术。如果你懂类似的算法并能实现之，恭喜你，你已经超越了 90% 的程序员了，你的重要性和价值将不可估量。

### （三）产品意识

做产品也是讲感觉的，当然有时我们觉得对产品的把握是产品经理的事儿，这么讲有时对，有时也不对。产品经理虽然是不挂衔的 CEO，但产品的最终实现，是要靠程序员的，假如你对产品没有热情、没有感觉，无所谓好无所谓坏，那很难做到出色了。

我有一个很朴素的想法。一个程序员，要想做一个不那么糟糕的产品，一定要有爱，就像天龙八部里的段誉哥哥，见到美女就爱得要死要活，对每一位妹妹的爱都是发自内心的、炽烈的、让人魂牵梦萦的。惟其如此，你才能像张京兆（张敞，曾任京兆尹）为妻子画眉那样用心雕琢你的产品，对每一个细节都念兹在兹直到完美。这是讲我们程序员要爱产品，爱各种各样的产品，这是做出好产品的前提。

你爱你做的产品，打心眼里希望它好，这就有了第一个前提。那接下来就很容易进入角色，每做一个产品都会有积累。慢慢你就会培养出对产品的独特感觉，形成产品意识，鹤立于程序员群体。

再啰嗦一点，业务积累是屠龙刀，技术积累是倚天剑，那产品意识就是张无忌了，由他来控制屠龙刀和倚天剑，就能发挥无上威力。

## 公司的招聘要求分析

在现实中，公司招聘程序员时，多数时候会提到业务和技术方面的任职要求，我截取了 two 则招聘信息，贴出来看看：

职位描述	公司介绍
<p>岗位职责：</p> <ol style="list-style-type: none"><li>1. 参与研发统一通信产品；</li><li>2. 负责即时通信及协作等Android上移动应用的分析、设计、编码、单元测试和集成测试；</li></ol> <p>职位要求：</p> <ol style="list-style-type: none"><li>1. 具备2年以上Android系统开发经验，完成过2个以上Android手机应用并成功上线；</li><li>2. 具备J2ME或JAVA开发经验，熟悉Android手机软件架构；</li><li>3. 对嵌入式应用特别是手持设备性能调优有实际经验；</li><li>4. 熟悉Android手机硬件结构者优先；</li><li>5. 研发过移动互联网即时通讯类产品者优先；</li><li>6. 思路清晰，思维敏捷，具备快速学习能力和良好的英文技术资料阅读能力。</li></ol>	

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

这是招聘 Android 开发工程师的，职位要求里的第 5 点，就是行业业务要求。1、2、3、4 都是技术要求。

再看一个：

### 职位描述

#### 工作职责

- 1.安全产品的底层驱动开发，实现浏览器保护、木马查杀等主动防御功能
- 2.竞品软件的驱动技术等关键技术的研究与分析
- 3.安全软件与Windows内核相关的技术调研工作，完成技术原理分析
- 4.安全软件Windows内核相关的功能编码实现与维护工作

#### 岗位要求

- 1.三年以上项目开发经验，三年以上Windows平台网络驱动和文件过滤驱动开发经验；

#### 岗位要求

- 2.精通C++，熟练运用STL，熟悉Windows驱动框架和原理
- 3.有应用层和驱动层Hook引擎开发经验（支持32位和64位）
- 4.熟悉各种DLL注入技术原理（支持win7,win8，支持32位和64位）
- 5.熟练使用OD、windbg、IDA pro等工具，有较强的程序反汇编技术和逆向分析能力
- 6.有信息安全相关项目开发经验优先

这个职位要求很硬很牛，大部分都是技术方面的，不过第 6 点，依然是和行业业务相关的。

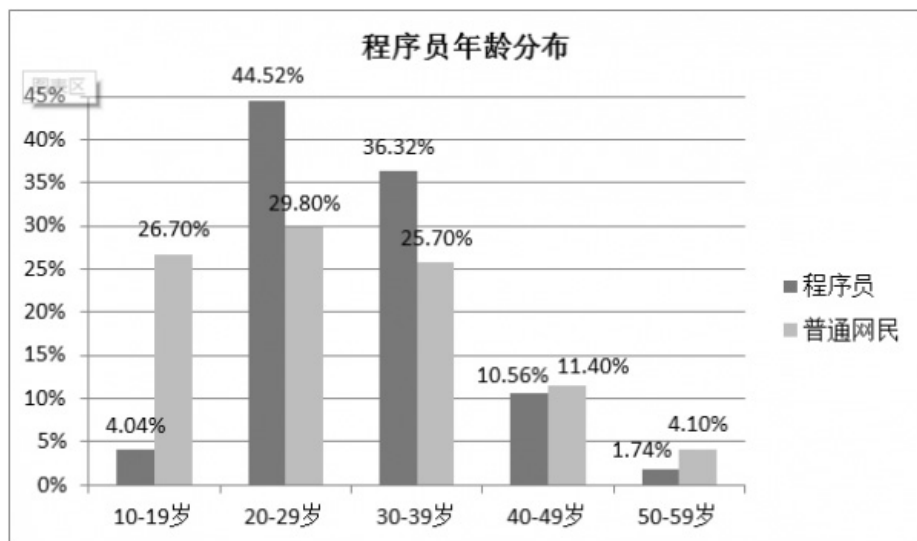
其实我比较中意的是看一个人是否喜欢、热爱他所做的事情，如果你爱，泛爱众，又是真爱，那你就有了坐着火箭飞行的基础了，如果没有这一点，顶多是做到称职而已。不过你翻遍招聘网站的招聘需求，这方面的要求都很少见到，原因大概是很难描述、量化，因为它实在是一个软指标。

## 结论

我要亮出我的观点，评说就任人了。

我的结论是：程序员的价值与年龄没有直接关系，价值不会反比于年龄。作为软件开发人员，自己不要给自己在年龄上设限，很多事情我们没做成，很多时候我们没能取得更大的成就，都是因为我们先给自己设置了限制，用这个限制禁锢了自己。

下面这张图来自 [cnbeta](#) 的文章“2014 年中国程序员调查报告”，它说明了国内程序员的年龄分布情况：



谁说程序员干不过三十岁呢，该图表明，30 岁以上的程序员占比高达 48.62%，占整个程序员群的将近一半。

2015 年 1 月 22 日，我去参加 Qt 西安开发应用研讨会，有一哥们儿就是老程序员，40 多了，宝刀正锋利呢，说起系统架构，说起各种技术，如数家珍熟之极矣。

王江民 30 多岁写软件，43 岁搞出横扫杀软市场的江民杀毒。

我的老朋友，40 来岁了写直播流媒体协议，P2P 的，传输效率高且稳定。

.....

☞ 你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

年龄只是一个自然现象，而我们的经验、认知、能力、心智成熟度，在合理的情况下，都是随着年龄增长的，廉颇不老，黄忠能战，只要你走在正确的方向上、有值得称道的积累，缘何不能横刀立马顾盼自雄耶？

## 国外的研究论文和结论

让我们看一下国外牛人的研究吧：“Is Programming Knowledge Related To Age?” (<http://people.engr.ncsu.edu/ermurph3/papers/msr13.pdf>)。这个链接是英文的，pdf 版本，基于 Stack Overflow 上的数据分析得出的结论。嫌 E 文难看的，来看个中文的：“编程能力与编程年龄的关系” (<http://dev.yesky.com/394/35423894.shtml>)，该文引用了 “Is Programming Knowledge Related To Age?” 的数据和结论。

“Is Programming Knowledge Related To Age?” 这篇论文的结论是：

- 程序员技术能力上升是可以到 50 岁或 60 岁的。
- 老程序员在获取新技术上的能力并不比年轻的程序员差。

随着年龄老去的，是我们的身体，而永葆年轻的，则是我们的心。心不老，青春在，宝刀寒光仍闪耀，依旧斩上将首级于万千敌阵中如儿戏尔。程序员，走自己的路，让用年龄唱衰我们的人说去吧。

## 不要让自己成为问题

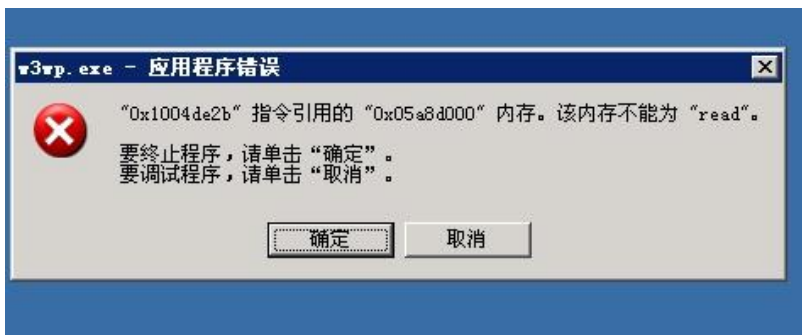
程序员（软件开发工程师）有很多别称，比如“程序猿”、“码农”、“代码工人”、“码字猴”等，都是幽默的“阿猿”们自“黑”的结果。这些个别称，指出了一个不争的事实：程序员要写代码。然而，被遮盖的问题是：程序员写代码的目的。

### 程序员的职责是解决问题

软件的目的是解决生活和生产过程中的问题，为人们带来便利，提高生产力和生产率。

程序员生产软件，所以，程序员的终极目的是用软件来解决自己和他人生活中的问题。这是程序员的职责，而写代码，不过是解决问题这个复杂旅程中的一小步，非常非常小的一步。

### 开发软件苦难重重



程序员是为解决问题而生的，可实际上，程序员的工作过程中，需要面对许许多多问题，问题让程序员痛苦，一个问题解决之后，新的问题马上又会出现，让程序员继续陷入无尽的痛苦。软件开发其实就是一个不断面对问题和解决问题的过程。所以，从这个角度来讲，开发软件苦难重重。

考虑软件开发的整个过程，大概有这么几类问题：



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

- 产品定义问题
- 架构设计问题
- 实现偏离需求
- 软件 Bug
- 维护问题
- 沟通问题

具体的问题我们也无须展开来说，不是本节的重点。本节的重点是，如何面对问题。首先我们要看看几种现实。

## 鸵鸟症



据说鸵鸟在遇到危险时会把头埋进沙子。这是逃避问题的一种方式。

很多程序员也采取类似的鸵鸟策略来逃避自己面临的问题和需要承受的痛苦，问题解决不了，就往后拖，当没看见。

这都是自欺欺人。虽然鸵鸟策略会让程序员暂时避免痛苦，但却不得不去承受另外两种更深邃的痛苦：看不到希望的交付期和没有尽头的维护。

## 贴膏药

还有一种常见的处理问题的策略，就是贴膏药。这种在解决软件 Bug 时比较常见。

说实话软件的质量和程序员个体关系太大，有追求的程序员会写简洁、有效、逻辑清晰的代码，出了问题比较好分析、解决；敷衍的呢，就随便搞搞能 Run 就行，出现 Bug 就很难解决，总是寻寻觅觅寻不到，最后呢，就只能是揣摩着在某条路上堵点东西，贴块膏药，久而久之，这代码就丑陋得不行，到处是狗皮膏药，你要是想揭开看看里面到底是什么，都无从下手。说什么飞越迷雾把代码看清楚，都是梦啊。

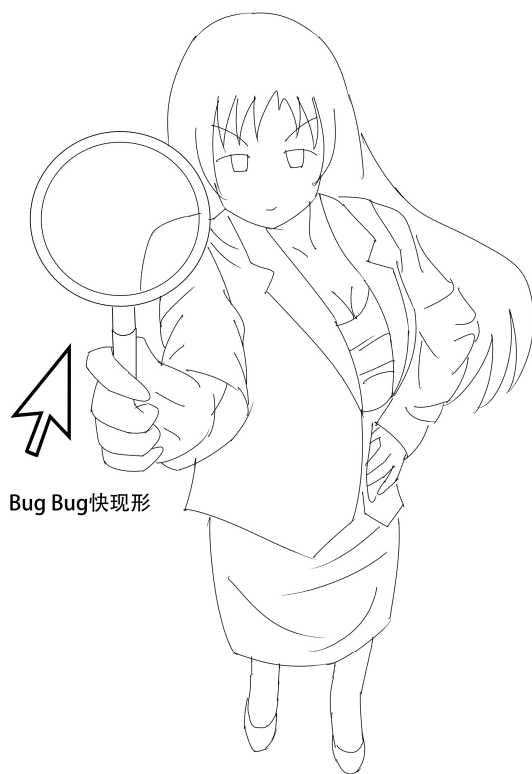
## 绑架产品经理

有一些问题来源于开发实现与产品需求定义之间的分歧。这个时候就看开发人员和产品经理沟通的结果了：要么产品妥协，把一个问题定义为使用习惯什么的以便延后解决；要么打回给开发重新实现。

我见过很多次妥协的情况，有的问题真的还蛮重要的。我认为这种情况，多数时候是程序员运用自己的技术优势为自己的理解偏颇保驾护航，最终绑架了产品经理。这是程序员不愿意看到、看到了也不愿意承认的事实。

🗨️ 你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 接受苦难，实现超越



令人欣慰的是，只要接受了软件开发苦难重重的事实，程序员就能从苦难中解脱出来，实现产品和自我的超越。程序员正是在承受痛苦和解决问题的过程中，技能、经验和心智得以成长和成熟。如果害怕面对问题，畏惧承受痛苦，一心想要逃避，那么我们虽然能够逃避一时的痛苦，却会承受随之而来的用户投诉、产品失败、公司破产等更严重的后果。

你不解决问题，你就会成为问题。作为程序员，这是我们一定要牢记的。

面对问题的泥沼，不可能跃过，我们的选择，只能是穿越。

## 直面问题的好处

程序员都是成年人了，应该有能力去面对问题和痛苦。但实际情况是，作为成年人的程序员，心理却不一定成熟，有时还是会像小孩子一样，不愿去面对和解决问题。这有能力问题，也有心理问题，但从旁观者看来，这归根结底是能力问题。

我们来给能力下个定义：解决问题的态度、技能、经验和自我学习的欲望，综合起来就是能力。

直面问题，会提升能力。这是最大的好处。

规避问题，会失去提升的机会，阻碍自我成长。但我们常常会这么做，一方面是我们倾向于认为正在做的事儿不是我们自己的，是公司的，老板的，我们还有退路；另一方面是因为接受问题、解决问题会挑战我们的技能、经验、认知习惯，是沉重而痛苦的，人的天性中有一点是拈轻怕重避重就轻，说白了就是愿意当下舒服。

软件开发过程中的问题不会因为程序员的回避而消失，问题会一直存在，并且老问题会滋生新问题，导致问题越来越多，超出产品、团队、公司所能接受的范围，最终在某个时刻集中爆发出来。

我有过很多次这样的经历，鸵鸟症让我不自觉地延后问题，总觉得最终交付期限还早，总抱着不切实际的幻想，自己骗自己说随着时间的推移也许问题会自己消失，或者出现一些情况让问题不再是问题。但实际情况是，项目不断延期，测试结论总是“达不到释放标准”，为了上线不断地加班，搞得大家伙又疲惫又无奈，期间的痛苦甚至让人绝望，我们都被问题淹没了，看不到出头之日。这个时候我是真心后悔没有及早暴露问题、解决问题，有时也告诫自己下一个版本、下一个产品一定不能这样。可事实是，恶性循环一旦形成，惯性很大，很难改变。这会让人厌烦、不堪忍受、不得安宁，努力多次后，当发现在当下的环境里无法改变这种恶性循环时，我会倾向于换个环境重新开始。这也是换工作的一种常见的原因。

为了避免这种情况，最好的办法就是克服就轻、逃避的欲望与习惯，接纳问题，拥抱痛苦，遇佛杀佛遇鬼杀鬼，第一时间解决各种问题。不要担心自己水平不够解决不了，人的能力都是在做超出自己能力的事情中提高的，这也是自我修炼的要诀：设定比自己能力高一级

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

的目标。

## 解决问题的实践

我们把心态放开了放平了，接下来就是如何解决问题了。每个程序员遇到的问题都不一样，解决问题的具体方法也天差地别，但还是有一些指导性的原则可供参考。

## 有效沟通

良好的沟通能力是优秀程序员最显著、最重要的特点。

如果你不能和别人有效地沟通，那你很难打磨出一个优秀的产品。

程序员需要和产品经理、市场、客户沟通需求、设计等问题，这是逃也逃不掉的。虽然有些程序员选择软件开发就是为了避免和人打交道，但残酷的事实是，在如今兵团作战的环境下，“两耳不闻窗外事，抱着键盘敲敲敲”的埋头苦干是做不出好东西来的。

程序员还要和开发同事、测试 MM 沟通。

如果我们能准确地表达我们的想法，正确理解别人的意图，那就可以在产品初期规避很多问题。

## 尽早暴露问题

问题暴露得越早越好，解决的成本也越低。如果临近交付时蹦出来神问题，会击垮可怜的程序员，他们会更倾向于回避、掩盖问题，有时实在没辙了，会和测试、产品一起，大家伙一起把自己说服，改变问题的属性、范畴、严重程度，然后相互安慰这看起来不是什么问题。

从软件开发过程来看，下面的做法会尽早暴露问题：

- 需求评审
- 设计评审
- 同级代码审查

不要让自己成为问题🐼

- 持续集成
- 迭代测试

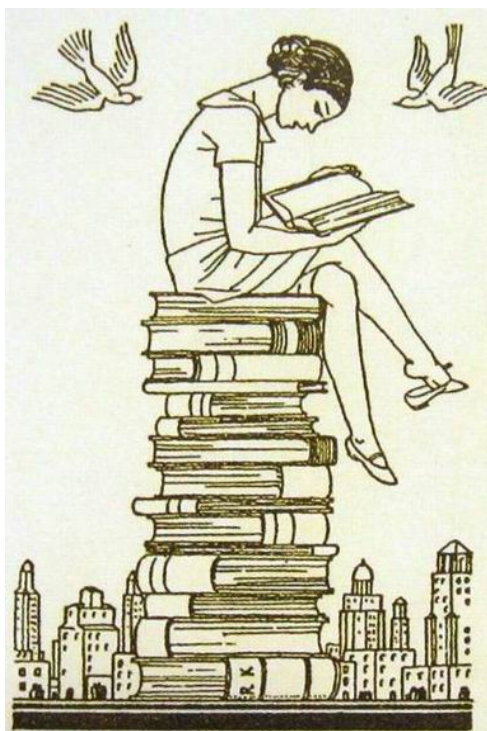
## 正确认识问题

无论怎样有效沟通，无论你采取多少措施来尽早暴露问题，还是会有预料不到的事情发生。这就是软件开发的不确定性，是我们必须接受的事实。而对于程序员来讲，一定要牢记，你在生产代码，也在生产 **Bug**。我们要正确认识这个事情，出现问题时，不轻视，不回避，不隐瞒，不悲观，实事求是地面对，实事求是地解决。

如果我们不能正确地认识问题，那迟早有一天，我们自己会成为问题，那时，我们面临的，将是被冷落、被淘汰。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 怎样才能多读一些书



本节我们来谈谈这个问题：怎样才能多读一些书？

### 你喜欢读书吗

我喜欢读书，看到书就喜欢，不读也要买，书柜放不下就堆床头，床头放不下就堆饭桌……当我满眼都是书时，即便不至于漫卷诗书喜欲狂，也会由衷感到读书是一种慰藉。

所以，决定你能否多读一些书的第一点就是你是否喜欢读书。

书不仅是生活，而且是现在、过去和未来文化生活的源泉。

如果你感到“人离开了书，如同离开空气一样不能生活”，如果你“三日不读书，便觉语言无味，面目可憎”，那怎么多读一些书对你来讲就不是任何问题。

那如果你对读书没有特别的感覺呢？

其实也还是会有机缘引领你去读书。

比如你喜欢软件开发，那你一定会去读编程的书籍。因为你喜欢编程，就渴望做好这件事，渴望成为佼佼者，渴望做出好的产品。前人的经验，一部分在你看不到的别的公司的产品里，一部分在公开发行的图书里，你会自然而然找到这些书，把读书当作挖矿，努力从沙里淘金丰富自己。

## 读书的益处





☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

也许说一些读书的好处，会让我们更愿意去读吧。

书籍是人类进步的阶梯，其实，读书就是将人类浓缩几千年的科技、文化快速习得的最佳方式，读书能够让你在极短的时间内，掌握大量的科学文化知识，突破沧海一粟的局限，将有限的自己融入到无限的世界里，增加自己人生的厚度和广度，让你的生活更加丰盈有趣。

快节奏的生活让程序员变成了陀螺，不得不在驯兽师的皮鞭下生活，一刻也不能停止旋转。读书正是对这种不由自主的生活的一种有效的调剂和治疗，让我们发现自己、认识自己、滋养自己，最终习得自己的节奏，走出自己的路。

读书有太多的好处，我们不能一一叙说，这里只结合程序员的生活特点，挑几样要紧的掰扯一下。

## 读书可以涵养静气

IT 江湖节奏越来越快，也越来越容易使人浮躁。

读书可以让我们撇开喧嚣，拨开冗务，在寂静中体会人生的滋味，在洗尽铅华的世界里观照自己，使人归于宁静和淡泊，使生命超然物外。

我的父亲曾经问我：读那么多闲书有什么用？

读书不像你饿了，吃个馒头就会饱；也不像你渴了，喝口水就会甘爽。读书的力量常常不能立即显现出来，也不能通过肉体感官体现出来。也正因如此，急躁的人，渴望立即反馈的人，总怀疑读书无用。其实，读书正像细雨和风，源源不断地潜移默化着我们，润物细无声。一个人的心灵若能得到知识的浸润，就会生出许多灵气和色彩。读书，是一种精神的跋涉，它能够涤荡我们的心灵，让我们的脚步从滚滚红尘的裹挟中挣脱出来，让我们重新学会放慢脚步，让心静下来。

浮躁的社会，心静者胜出，且从读书修炼，让我们看尽人间兴废事，行到水穷处，怡然坐看云起。

## 读书可以正三观

想想看，你是否努力活在别人的看法里？

想想看，你是否在跳来跳去追求加薪的运动中迷失了自己？

想想看，你是否越来越不明白要走向何方，徘徊在各种各样的十字路口，拔剑四顾心茫然，眉头紧锁空太息？

想想看，你是否以为不差钱了就会快乐，于是不断地拼命赚钱结果却越来越不快乐？

……

三观不正，身若浮萍，只能随风而动，顺水飘零。

胸无江海心难阔，腹有诗书气自华。

书中有着广阔的世界，书中有着永世不朽的精神，几千年沧海桑田，数百世物换星移，书籍永远葆有它独特的魔力，能够与相逢的每一个人碰撞出火花。

通过阅读，我们能够体悟先贤们的哲思，能够与文人骚客们煮酒论歌，能够从无数发人深省的故事中，吸取教训，增长见识，去粗取精，练就自己的定海神针。

读书，足以怡情，足以博采，足以长才，足以让我们得自在，足以让我们明是非，足以让我们穿越四季的变换，足以让我们发现自己完善自己，走出自己的心灵之旅。

## 读书可以提升才华

孙洙在《唐诗三百首序》中言：“熟读唐诗三百首，不会写诗也会偷”。

杜甫的诗：“读书破万卷，下笔如有神”。

刘向的话：“书犹药也，善读之可以医愚”。

培根的话：“读书使人充实，讨论使人机智，笔记使人准确，读史使人明智，读诗使人灵秀，数学使人周密，科学使人深刻，伦理使人庄重，逻辑修辞使人善辩。凡有所学，皆成

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

性格”。

……

多智而近妖的诸葛亮也说：“非学无以广才，非志无以成学”。

书是人类智慧的结晶，是人类进步的阶梯，读一本好书，就是和许多高尚的人谈话。书中自有黄金屋，书中自有颜如玉，书是文化、经验、知识的载体，是大千世界的浓缩，观之可以让囿居生活一隅的我们肋生双翅，超越身体、时间、空间的局限，无拘无束地畅游古今中外。

从小处说，读书可以丰富词汇，增加知识，积累谈资。腹有诗书气自华，程序员，你尤其需要这个，当你读了足够多的书，自然可以在和人聊天时旁征博引信手拈来，你会因此而更能说，更会说，更有机会找到女朋友。读书可以帮助程序员摆脱“人才不一定有口才”的窘境，让你进入“有口才的一定是人才”的行列。

从大处说，读书可以让我们明辨是非，胸有主见，不断完善自我。我们的内心世界将因读书而更加广博，我们的生活将因读书而更有厚度，我们的时间将因读书而走向无限。

书读得多了，知道的事情就多，思路就会变得更为开阔，解决问题的能力也会随之提升。

阅读的最大理由是想摆脱平庸，早一天就多一份人生的精彩，迟一天就多一天平庸的困扰。

除掉睡眠，人的一辈子只有一万多天。人与人的不同在于：你是真的活了一万多天，还是仅仅生活了一天，却重复了一万多次。而读书，正是我们摆脱日复一日平庸生活的开始。

## 时间从何处来

程序员，九小时在单位，两小时在路上，八小时在床上，两小时在吃饭……于是，每天就剩三个小时了，还要洗漱、上厕所，偶尔理个发买个衣服搞个社交……

这还是不加班的情况，假如遇上加班，天天晚上吭哧吭哧工作到十点以后。

这么看来，哪儿有时间看书呢？

其实，时间就像海绵里的水，只要你愿意挤，总是有的。虽然这话说烂了，但说得真心不错。

假如你原本是早上八点起床，如果改为六点，这就多了两个小时。

假如你原本每天晚上看四十分钟视频，如果戒掉，这就多了四十分钟。

假如你原本每个工作日中午休息半个小时，看看新闻，看看视频，看看漫画，打打游戏……如果取消，这就多了半个小时。

就这几项，是不是每天就有三个多小时可以看书了？假如你早上六点起不来，改为七点起床，每天也有两个多小时可以看书。如果你能坚持下来，一年三百六十五天可以看很多书，五十本我估计没有问题。

这还没算周末，周末两天，若是抽出四个小时读小说，估计可以读三十万字。

“三更灯火五更鸣，正是男儿读书时，黑发不知勤学早，白发方悔读书迟”。

这是颜真卿的诗，正可以指导我们怎样挤出读书的时间。书山有路勤为径，除此无他。

伟大的成绩和辛勤劳动是成正比例的，有一分劳动就有一分收获，日积月累，从少到多，奇迹就可以创造出来。就让我们从现在起，踏上我们的读书之旅。

## 伤心小箭，你中了几枝

程序员怎样说话才能让别人觉得舒服呢？这是个问题，我一时半会儿还搞不定它，因为我本身也不是一个“会说话”的程序员。我们换个角度来问，作为一个程序员，你不能接受哪些话？

### 否定性语言

有一个寓言，是这样的：

在一座大山的山脚下，住着一群山民，他们日出而作日落而息辛苦劳作却收获甚少。有一天，从山外来了一个游客，他对山民们说：“我知道一个咒语，只要对着大山，在心里默念这个咒语，你们面前就会出现一大堆金币。我可以把咒语教给你们，但你们要先给我一笔钱作为酬劳。”于是山民好不容易凑齐了那笔钱交给他，然后席地而坐洗耳恭听。山外来客开始传授：“咒语是：\*\*\*\*\*，但你们在念这个咒语的时候心里千万不能想到喜马拉雅山的猴子，否则咒语就会失效。”山民们说：“一定，一定。谁会去想什么喜马拉雅山的猴子呀！”

游客走了，山民作好准备，每个人都面对大山虔诚地念起了咒语。但是，那只该死的喜马拉雅山的猴子总是出现在脑中，他们愈是提醒自己不要去想，猴子的影象愈是清晰。

最终，没有人面前出现金币。山民们想：他说得没错，咒语果然失效了。

“喜马拉雅山的猴子”有很多寓意，我关注的是山外来客对山民们的特别提示：**千万不能想到喜马拉雅山的猴子**。“千万不能”这个词儿，是一个否定性词语，它使得这句话成为否定性指令。

我们的大脑更善于接受具体的内容，比如视觉、动作、听觉，像“不能”、“不要”、

“别”等等否定性的词语，属于语言文字层面的东西，没有与之对应的具体形象，我们在说它们的时候，听的人脑子里找不到这些词儿对应的形象，经常会不自觉地忽略掉它，所以，“千万不能想到喜马拉雅山的猴子”会激活“喜马拉雅山的猴子”这个形象，“别去想那只粉红色的大象”会激活“粉红色的大象”这个形象，最终听我们这么说的人，都做出了与指令相反的行动。

简单点儿说，就是你越不让某个人怎么怎么着，他越容易那么着。比如你告诉孩子别摸插座，他十有八九会去摸；比如孩子吃饭时你告诉他别把饭撒桌子上，他十有八九会把饭撒桌子上；比如孩子蹒跚学步，你告诉他别撞到桌子，他十有八九会撞到桌子；比如你告诉孩子别去厨房，他十有八九会去厨房看看……

其实，我只是在说一个事实：否定性词语更难被接受，否定性指令往往会适得其反。

上面提到的这些例子，还算是好的，毕竟它们仅仅是以否定的形式陈述一件事情，比它们更糟糕的场景还有很多，带来的效果则更让人难以接受。我们先看看常见的否定性词语以及在特定语境下产生“否定”含义的辅助词，再结合程序员的日常生活来看看它们的巨大破坏力。

## 常用于否定的词语

太多了，我简单列几个：

- |      |        |      |
|------|--------|------|
| • 别  | • 都    | • 不会 |
| • 不  | • 能不能别 | • 完全 |
| • 不能 | • 老这样  | • 没用 |

可能单看这些词没什么感觉，接下来我们结合程序员的工作与生活来看看它们是怎么被使用的。

你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 那些伤人话



在一个团队中，语言沟通是时时刻刻存在的，话说得是否得体，是否让别人听着舒服，在很大程度上决定了协作的效率。随随便便的一句话，看起来无伤大雅，也可能瞬间在两个人之间竖立起坚不可破的屏障。

每个人都希望得到别人的肯定，没有人总期待别人批评、讽刺、挖苦、呵斥、惩罚自己。可实际上我们经常忘记这个，脱口而出的话就带了讽刺、挖苦、呵斥、批评等味道，仿佛只有这样够严厉、痛快、解气，才能达到强调与教育的目的，其实往往适得其反。

我回顾自己的程序员生涯，对那些让我们不舒服、难受、憋屈、窝火、操气、愤怒、难

堪的话语做了简要分析与梳理，分出了下面几类：

- 否定能力
- 否定努力与结果
- 否定人
- 侮辱
- 命令

举例如下。

“这都不会？”

“这么简单的事儿你居然花了这么长时间！”

“你怎么总是搞错！”

“一点儿责任心都没有！”

“这么简单我一个小时就能搞定！”

“你写的这些完全没用！”

“这是一个很简单的功能，你用 20 分钟应该就能写出来。”

“别人都做出来了，你为什么说没办法实现？”

“就你的 Bug 多！”

“每次 Build 都得等你！”

“一行代码十个 Bug，能不能用点儿心？！”

“这代码写得太差劲了，简直没法想象！”

“这个 Bug 怎么还没解决？”

.....

可以设想别人对自己说这些话时自己是什么感受，我是会有受伤的感觉。



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

我想这些话，还有类似的很多话，应该成为禁语，它们除了伤害别人、制造麻烦，再没有其他作用了。不要让它们伤害我们以及我们辛苦工作、勤勉上进的小伙伴们。

## 避免说禁语



怎样才能避免说禁语呢？

有个词儿叫做“共情”，指的是一种能深入他人主观世界，了解其感受的能力。每当我们冲口而出某句禁语时，我们都要想想对方听了这话是什么反应，这话会对别人造成多大伤害，如果我们能感同身受，就不会再说这样的话。这也是“己所不欲，勿施于人”。

不过有些禁语很随意，说起来感到很痛快，常常会自行蹿出去，甚至让说者不觉。这就是否定性语言的“魅力”，它会让你感到一种力量，并用这种力量来迷惑你，给你带来一泻千里的瞬时快感。怎么办呢？

先冷静三秒！

当你要评价别人时，先数三个数，让自己冷静下来。一旦你打断自己评价的冲动，就有

伤心小箭，你中了几枝🔪

时间考虑要说的话是否合适了。

古典武装情景喜剧《武林外传》里的郭芙蓉有句口头禅——“世界如此美妙，我却如此暴躁，这样不好，不好”，如果我们能这样自省，也有助于远离禁语。

每次说话都考虑这么多，岂不太累？

如果你经常说这样的话，要改掉这个习惯，开始时是比较困难，需要经常提醒自己，刻意地约束自己，但做的次数多了，就会形成习惯，再往后就会成为你的一种行为模式，告别刻意转为自然，一切的美好，就会由此开始。

你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 什么仇什么怨



程序员的沟通问题，从我自己的经验来看，是个大问题，却也是最容易被不善言辞的“程序猿”们忽略的问题。

沟通的目的是为了解决问题，让事情朝着你希望的方向发展。但每个人都是一个谜，谁也无法解开的谜，你和我，他和她，常常是盈盈一水间脉脉不得语，甚至击键之声相闻而老

什么仇什么怨🐼

死不相往来。

两个人和一个目标，这三者之间就是个迷宫，怎样辗转腾挪高飞低伏，穿越重重迷雾，避过隐身暗处伺机要吞噬你的怪物，与那一端的那个她相逢一笑携手同行，这是比哥德巴赫猜想还要困难的问题。

千万别惹程序员

有胆再说一句！



不要以为程序员都是呆板的、面无表情的猴子，其实他们内心深处充满一种不愿为人所知的热情，也有一块虽不可见却极力保护的不可侵犯的圣地，要是你不小心惹毛了他们，轻

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

则攻击你的电脑，重则和你干上一架。

我采访了几个打过架的“程序猿”，下面咱来讲讲他们的故事，为了有代入感，我选择了第一人称，这是为了叙述方便，不要误会，笔者是斯文人，文弱书生，手无缚鸡之力，不会打架的。

在叙述的过程中，我可能会插入评论和解说，混淆故事角色与讲述人，再次强调一下，不要误解。如果你发现自己竟然是故事的主角，也不要自作多情，因为，无巧不成书，这世上类似的事情实在是太多了。

## 都是联调惹的祸吗

对我而言，其实那一场架都不知道怎么打起来的，也是我软件开发生涯里唯一一次因为工作而打架。

假设对方叫 J 吧，希望他看到不要介意，事情已经过去很多年了，再大的怨也该随着时间而消逝了。

当时我在做车载影音系统，用 Windows CE 系统，实现 DVD、多媒体、收音机、导航等基本功能，另外还有写流式驱动控制马达和电机来翻转屏幕（一开始设备是收在汽车中控盘面板里的，想象一下光驱）。当唤醒设备后，驱动通过 I2C 控制马达，先把盖子送出来，然后屏幕上翻，再然后就可以通过蓝牙或者手指触摸来控制设备。我当时做界面和流式驱动，而 Windows CE 的 SDK 的一部分由 J 负责。

问题就出在接口上了。

项目很急，客户急着要去参加 CES 电子展，我和硬件工程师不久就要现场开发，而驱动还没搞定，我就去找 J，问他什么时候可以联调。

联调可能很花时间，有时各自写代码花不了两天，一集成一联调就出现问题了，写两天代码联调一星期的事儿都可能发生，这也是敏捷为什么强调持续集成的一个原因，大爆炸式的广度集成，会带来意想不到的悲剧，各种突然与惊吓，真是“银瓶乍破水浆迸，铁骑突出刀枪鸣”，大家都会被各种突发状况给搞得头懵脑胀，很多时候只能停杯投

箸不能食，拔剑四顾心茫然，到最后甚至一夜白头，或者经过数夜的煎熬开始向吸血鬼进化。

我找 J 联调，J 当时不知道在忙什么，不搭理我，头也不抬。我就和他说我很急，J 回头表示他没时间。可是我真的不知道他在搞什么，我就强调项目的急迫性。几个来回下来他首先说脏字骂人，我肚子里正窝着火呢，一听这厮居然还骂人，就和他打了起来。我一把就把他摁倒在办公桌上，我俩就当着办公室的同事的面开打了。不过说实话，我这人心慈手软，虽然摁倒了 J，也没怎么打，反倒被他用指甲在脸上抓了几个血印子。

这种事儿自然会被领导知道，后来 J 被调到另一个部门，再后来 J 离职了。再再后来，我也离职了。

其实我真不知道我为什么会因为这些鸡毛蒜皮的小事和 J 干架。我的大学同学曾经说我干出什么事都不奇怪，不幸被他言中了。

在打架之后的一个星期或者更长时间内，我都处在一种异常的状态里，触景生情，一出家门或者到办公室，就会想起来，心跳会加速，面红耳热，我并不为当时的冲动懊悔，做了就是做了，但冲动终究是会收获惩罚的。

小孩子才问“你为什么不理我了，是不是生我气了”，成年人都是默默地相互疏远，哪怕只是一点点嫌隙，也可能发展成永生无法填补的深渊，最终相互之间形同陌路。

因此，人们常说，多个朋友多条路，结个冤家多堵墙。

可是，我们都是受情绪支配的猴子。

## 乱斗的，不只是西游

首先声明，这是一个故事，如有雷同，纯属巧合。

笔者和贾雨村是聊得来的朋友，下面的故事是我采访贾雨村后整理出来的，为了方便讲述，有时采用第一人称的叙述视角，不要误解。

贾雨村开讲了：

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

“我在一个小小的软件公司当部门经理，好吧，我不懂技术，我只是个管人的，技术与我何干哉，我和老板是朋友，这个部门经理是老板友情赞助我的，他说你不用懂技术，管好人就行了，我天真地以为这是真的，不懂技术也可以管那帮技术猴子！我根本不懂这里的‘程序猿’们在使用什么语言、什么框架、什么工具，以及他们的软件是怎么开发出来的。所以，我好失败。但这些，都不是最要命的，最要命的是，老板给我的这个部门总是不能按时交付，延期延期再延期，最后勉强交付还存在一堆问题。我背负了改变这一现状的伟大使命，因为老板说造成这一切的不是技术问题而是人的问题，所以，我不懂技术也没关系，拿住人就能拿住事儿。我很天真，就答应了朋友。于是我只能从项目的角度去看待，这里真是一片大有可为的好天地。不幸的是，我不想看到的事情发生了……”

笔者在“程序员的生活就这样吗”一节中讲了一个发版本的故事，这里是另外一个故事，依然是发版本。

贾雨村他们计划 2014 年 12 月份发布一个电商网站，其实他们已经有一个线上运行的购物平台，只不过有点儿小问题，有时会删掉卖家自定义的 SKU，有时会搞错促销打折的日期，偶尔也会算错钱多扣买家几块……所以使用他们软件的合作伙伴怨言很重怨气冲天，强硬地要求他们在元旦前更新版本，否则就拒付 30% 的合同款，还要告他们，让他们赔偿损失。

老板脸都绿了，恶向胆边生，决定在某个夜黑风高的夜晚潜进这帮程序员的家里，提刀砍掉这帮家伙的头，就像切西瓜一样，切完了还要摔到地上，让汁液横流，然后再狠狠地踩上几脚。不过因为现在是法制社会，要加强法制建设，依法治国，老板不敢顶风作案，只好硬着头皮到客户那里领了一河滩骂回来，值得庆幸的是，客户给了他们两个月的时间，限期整改，12 月 31 日必须上线，元旦铁定要旧貌换新颜。

于是，强悍的“程序猿”和富有经验的“产品狗”决定来一次华丽丽的转身，闷头搞上两个月，彻底扭转他们在客户心目中的不良印象。

可有句话是这么说的：要改变你留给别人的糟糕印象，有时可能需要一生的努力。其实也很好理解，一死恩仇泯，谁会和死人较劲呢。这句话反过来的意思是，如果你把事情搞砸了，给别人留下了非常糟糕的印象，除了死，是没有办法洗刷你的清白的。所以，你看很多电影都

用这种桥段，以死明志。

程序员的世界当然没有这么残酷，顶多老板因此失去客户或者破产，而你仅仅是再去换一份工作，再搞垮一个老板而已。

贾雨村觉得这种封闭开发不靠谱，因为这种做法无异于刻舟求剑，客户的需求是随着时间的变化而变化的，一如在水上滑行的木舟，而亲爱的程序员们却异想天开地认为冻结住需求（在木舟上刻个记号）吭哧吭哧猛搞两个月，就可以交出一个漂亮的电子商务网站。这不是痴人说梦吗？

让我们把时间定格在 2014 年 12 月 20 日（周六），这是一个风和日丽的早晨，帝都终于又拥有了一个 APEC 蓝，大家上班的时候虽然有些同情帝都方圆几百里不能烧火做饭的居民，可这轻微的忧思很快就被罕见的 APEC 蓝带来的巨大喜悦给掩埋了，直到大家打卡进入办公室，才从幻梦中醒来：原来无论 APEC 蓝多蓝，我们这帮“程序猿”还是要开着日光灯在密不透风的办公室里写程序！还不如那些不能烧火做饭的帝都近郊的居民。

还没等这种忽上忽下阴晴不定的情绪波动平稳下来，贾雨村又叫大家开会讨论进度了。

事后被采访的猿五先生表情迷离，抛出了几个谁也给出答案的问题之后，又转身敲代码去了，因为蔡大说，你们的版本根本就没做完！

其实，“根本就没做完”是动手的信号，不过我们要放到后面去说。

贾雨村说：大家判断一下，12 月 31 日能不能上线？猿大，你说说。

猿大：我觉得没问题，功能都做完了，兄弟们也都完成了单元测试，还交叉测试了几次，整体表现比线上的版本强太多了，这次绝对让客户满意。

贾雨村说：蔡大，你看呢？

蔡大：从测试的角度看，我们一轮完整的测试还没完成，要完成第一轮测试，预计还需要 10 天。而根据以往的经验，一轮测试根本不可能达到发布标准。

忽然之间，贾雨村的脸蒙上了黑纱，就像初入江湖的圣姑（事见《笑傲江湖》）一样，不过贾雨村远没有圣姑那么迷人罢了，他发出的不是吸引异性的气体，而是难以掩盖的火药



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

味，只是现在还在沉默而已。

猿大：BOSS，从开发的角度讲，功能都完成了，我有把握 12 月 31 日可以上线。

贾雨村：……

蔡大：还没测完一轮，还不知道有多少问题没暴露出来，我觉得肯定不能上线。

猿大：我们已经测过了，你们测了这么久了才出现了 200 多个 Bug，我们修复得也很快，才几天就七七八八了，对吧，我们修复 Bug 的速度比以前快多了，肯定没问题。

贾雨村见气氛不对：其他人都散了吧，猿大、蔡大，还有品大，你们三个留下。

贾雨村：到底能不能上？品大，你看过系统没，与需求有没有什么出入？

品大：大概看了一下，出入不大，只有十来个功能不一致。

蔡大：其实很多地方，开发实现的都和需求不一致，而我们测试根本不知道这些修改是什么时候定的，问品大，品大又不配合。

品大：你怎么说话呢，谁不配合！你给我说清楚！

蔡大：我说的是事实，测试的小姑娘们问你需求问题，你总是不耐烦，说话很不客气，大家都不敢问你了。

品大：我有吗？再说你们问的问题需求文档中都写了，我哪有时间一次一次纠缠这些细节。

蔡大：需求文档和开发实现的系统对不上的地方太多了！

猿大一直在旁边憋着劲儿，两眼罩住品大，就差冒烟了：居然敢说有十几个功能和需求不一致！等待品大和蔡大的争执有了缝隙，猿大马上接过话头。

猿大：品大你刚说有十几个功能和需求不符，不能这么做人吧，实现成什么样子都和你对过的，怎么不早说！

品大：昨天才对了一次，之前从来没给我看过！

猿大：你敢说之前没给你看过？

品大：你愿怎么说怎么说吧，总这样！

猿大：我们在邮件组里都说了，品大，蔡大，就你们俩，老不关注，11月我们就提供了测试版本，蔡大你们到上周才开始测，要是早点儿开始的话……

蔡大：11月22日的版本连冒烟都通不过。

猿大：我们都自测过的，你别乱说话，真的是哪里都不能点吗，就没一个能点的功能吗？

蔡大：购物车没办法使用，很多流……

猿大：我让你测购物车了吗？我们告诉你购物车功能完成了吗？那个版本就是商品展示那些页面，你们……

蔡大：你们根本没有告诉我们测试范围！

猿大：不可能！送测说明里都写了！

贾雨村饶有意味地看着几个人脸上慢慢浮现的青筋，听着他们越来越高的声音，他好长一段时间没有说话了。现在他说：到底能不能如期上线？

猿大拍胸脯说没问题，蔡大斩钉截铁说不可能，而品大陷入了恍惚，脸上的表情就像天上的云，在风吹之下倏忽来去。

贾雨村提高了声音，再次说：到底能不能如期上线？

蔡大说：我先说说我们测试的情况，测试的开始时间是12月15日，这么短的时间这么多功能，到现在也没覆盖全，完成度在30%，不可能在12月31日前测完。由于留给测试的时间太短，如果到时不能上线，我们测试人员不承担责任。

猿大：12月15日？11月22日版本就提交了，12月1日还提交了一个版本，怎么你们是从12月15日算？

蔡大：那两个版本都不能用，冒烟都没通过，根本不知道做的什么。

☞ 你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

猿大：别老跟我提通不过！通得过通不过你说了算？

蔡大：冒烟不通过就是测试说了算！

猿大：跟你没法说，BOSS，没事儿我出去了。

贾雨村赶忙说话：等等，还没讨论完，到底能不能如期上线？

蔡大：到现在功能都没做完。

信号就是在这这一刻发出的，“没做完”，这就是动手的信号。

请记住，大家是站着开会的。贾雨村管这个叫作“站会”，这是敏捷开发中非常看重的一个实践，有人也叫它每日立会。站会有很多好处：

- 让大家尽快投入到一天的工作中来。
- 如果某个开发人员在某一点上有问题，他可以趁此机会公开问题寻求帮助。
- 项目经理或管理层可以了解到哪些领域需要帮助，可以据此重新分配人手。
- 团队成员都可以了解整个项目的进展，彼此形成激励。
- 帮助团队识别重复劳动。

.....

可是，现在出问题了，就因为是在站着的，更方便动手——“没做完”就是信号，说时迟那时快，猿大右手一把抓住白板擦就甩向蔡大，左手戟指蔡大，嘴里高叫着“你再说我们没做完，你再说一遍！我们天天晚上加班到十二点，你天天一下班就溜，你还敢说我们没做完！再说一遍试试！”。

蔡大也不能示弱啊，都是男人，哪有认怂的道理！蔡大抬胳膊挡住白板擦，把手里的 Notebook 和中性笔用力掼到地上，圆睁怒目，高叫回应：“怎么啦，没做完还不让人说啦！”。

“没做完”，信号再次发出，猿大张牙舞爪向蔡大冲过去。贾雨村和品大赶紧拉架，品大拉猿大，贾雨村拉蔡大。

品大太瘦小，猿大身高一米八几体重将近 200 斤，冲劲儿很足，品大有点儿拉不住，

正要抱腰呢，猿大飞起一脚就踢品大肚子，嘴里骂骂咧咧的：真有十几个功能不对，真  
有吗？说话不问良心！品大闪个身，大腿挨了一脚，脸瞬间黑了，嘴里还击：猿大你不  
知好歹，疯啦，谁都打。猿大不理品大，朝蔡大冲过去玩起了“无影脚”和“化骨绵掌”，  
连贾雨村都中了几招。贾雨村上折下挡，嘴里高叫“你们这是干什么！”，心里怒火中烧  
几乎不能自持。品大个子虽小，也是个有性子的人，跟在猿大后面也要踢猿大。可怜贾  
雨村一个人，胳膊腿儿都用上，也遮挡不完，一不小心，品大和蔡大不知怎么也干起  
来了。

这时，整个公司的人都被惊动了，一众“程序猿”和“攻城狮”速度围了上来，七手八  
脚地，费了九牛二虎之力，终于把三只“愤怒的小鸟”拉扯开。

公司的天条中有一条：办公场所内打架斗殴者，杀无赦。

那么，到此结束了，这个故事，后面的不需要讲了。

你知道吗？贾雨村当时怒火中烧几乎不能自持，有那么一刹那，他真想加入战团打个高  
兴，脑海里已经出现四方混战的火爆场面了，手都激动得发抖了。

## 和程序员沟通很难吗

拳头是最无力的沟通方式，这与它所代表的肢体上的强力恰恰是相反的。所以，前面打  
架的事情，都是反例，忘了吧。现在让我们专注本文的中心主题——沟通。

下面解释两个名词，有助于我们理解程序员面对的沟通问题。

**沟通：**指为达到一定目的，将信息、思想和情感传送给对方，并期望得到对方做出相应  
反应效果的过程。

**沟通障碍：**指信息在传递和交换过程中，由于信息意图受到干扰或误解而导致沟通失真  
的现象。

我们先来看看程序员的一些特点及其与沟通的关系。

你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 程序员的特点



我这里说的是一部分程序员的特点，你没这些特点，你周围的人可能有，你就面临与具有这些特点的人沟通的问题。反之亦然。

## 喜欢以逻辑上说服人

如果逻辑思维能力比较差劲，基本上很难成为一个优秀的程序员。而有追求的程序员通常都有比较强的逻辑思维和抽象能力。这当然是非常好的特点，可是因为这些程序员天天和电脑、代码、设计等打交道，而这些对象又具有鲜明的确定性，逻辑正确基本上结果就会正确。开发人员的一大快乐就在于此，想到了就可以做到，就会有成就感。

人是什么样子的呢？如果做一件事情有好的结果，有成就感，能够得到肯定，就会倾向

于频繁地做类似的事情，强化既有的经验，最终形成一种定势。

程序员能够从开发工作中获取到快乐，他就会强化这种因为逻辑而来的快乐，而软件这玩意儿头脑又比较简单，你说啥他听啥，反馈到程序员身上就会帮助程序员形成简单、直接的思维定势，认为只要逻辑正确、道理明白，就应该得到应有的反馈。

其实这种认知原本没什么错，只不过，人不是电脑不是软件，人有色、受、想、形、识五蕴，有眼、耳、鼻、舌、身、意六识，普通人又不可能五蕴皆空，个人习惯、性格、情绪时时刻刻都在影响一个人对信息的接收、处理和反馈。并且，每个人都是不一样的。程序员因为收拾代码习惯了，会不自觉地将相关经验应用到与人的交往中，就很容易出问题。

问题的根源就是重逻辑、重告知，忽略情感、轻视反馈。

太枯燥了，我们举个例子。

比如有的程序员发现另一个“猿友”的代码有问题时经常会直接指出来，说“那谁，你这块代码逻辑不对”，或者说“那谁，你这代码根本就编译不过去”，或者说“那谁，你的代码一运行就崩溃”，凡此种种话语，你留心观察的话，经常会听到。这种话很直接，很明确，逻辑上正确。可是有时就会惹恼人，因为有的人就是会面子上挂不住，觉得你在挑他的刺儿，甚至觉得你在表示你比他能干，表示你在鄙视他。

就这样几句话，如果是说给我听的，我听到 100 次，99 次会过去和说这话的人讨论到底是什么问题。也有的程序员听见这话就来气，下意识地反驳是常有的事儿，连反唇相讥都有可能。人和人是不一样的，都有自己的话语过滤器，同样的话经过不同的过滤器，进入到这个人心里的時候，产生的效果就大不一样。所以，有时张三听了一句话会哭，李四听了同样的一句话却会笑，就是这个道理。而我们这些“程序猿”，有时是不能想象这种简单事实的，或者可以想象却被习惯性忽略。

## 我执

我执，究竟是什么呢？

用简单的话说，我执就是对一切有形或无形事物的执着。我执是一切痛苦的根源。如果

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

你不在乎一个事儿，那别人拿这事儿就刺激不了你。比方说你对男女情爱毫不在意，那别人抢走你的女朋友你觉得也没什么大不了。就这么简单。

佛法说：“世间诸灾害，怖畏及众生，悉由我执生，留彼何所为？”

你看，佛教修行的一个重点就是消除我执。而我们普通人基本上天天都在我执的痛苦里挣扎。

程序员有自己执着的東西，有一部分程序员很在意自己的劳动成果，对代码所有权非常看重，讨厌别人修改他的代码，讨厌别人说他的代码不好，讨厌别人说他的程序不好，进而讨厌那些说他代码、程序不好的人。就像上文中第二个打架的故事那样。

对于这样的一些程序员，代码比他们的脸还重要，你说他的代码有问题就是在打他的脸。如果你还是不拐弯地说他的代码不好，那就是猝不及防地、狠狠地抽他的脸。那么他怎么能够接受呢？怎么会没情绪呢？怎么会和你好好地讨论如何解决问题呢？

## 相轻

有个词儿叫“文人相轻”，其实程序员也是一样的。如果一个人特别牛，“阿猿”难望其项背，那也没什么别的念想，感叹一下而已，转回身就忙自己的事了。如果感觉一个人各方面能力和自己差不多，工作上却能比自己受重视，那“阿猿”就会不服气不平衡，就会觉得那谁谁和我差不多，为什么领导总是表扬他？种种不平就这么产生了，如此一来两个人之间的沟通也就可想而知。

## 推己及人

迈克尔·杰克逊说：别站在你的角度看我，你看不懂。

所以有时我们就很希望能情景互换，换位思考，把自己调换到对方的位置来想问题、理解问题。

很多人、事、情会超出我们的理解能力，我们不可能真的理解，哪怕你虚怀若谷、忘却自我，努力地把自己想象成另外一个人。

但我们都在努力，也一直被教育要努力理解别人，说什么人同此心心同此理。

有个词儿叫作“同理心”，还有个词儿叫作“情景代入”。

当一个人遇到了糟心事儿，比如男朋友丢了，找你来倾诉，你可能会说：阿媛，我能理解你的痛苦。但是，你真的能理解吗？

我发现自己说话和写博客的一大毛病是下笔千言离题万里，不过我也不打算改变这个。

那么，问题来了，我到底要说什么呢？

我要说的是：多数时候我们在理解、感知、评说一件事时，是从自己的经验出发的，这是必然。这就是推己及人。

抱歉，程序员尤其容易这样做。这是因为他们的工作习惯强化了他们的思维习惯，最终导致他们推己及人推得厉害。

那么推己及人不好吗？人家说，己所不欲勿施于人，挺好的啊。当然这是好的一面，不好的一面是：

你觉得自己好好学习天天向上所以别人也是如此；你觉得自己会虚心接受批评所以别人也是如此；你觉得自己为了项目目标应该努力拼搏所以别人也是如此；你觉得自己上班不打游戏所以别人也是如此；你觉得自己加班干活所以别人也是如此；你觉得自己老逛淘宝所以别人也是如此；你觉得自己屏幕上老是广告乱飞所以别人也是如此；你觉得自己周一周二进不了工作状态所以周四又开始等周末所以别人也是如此；你觉得拿一分钱干一分活，项目成功或者失败与你没有关系所以别人也是如此；

其实不是！

你犯了以己度人、以自己为标准要求别人的错误。请不要这样，别人之所以是别人，就是因为他们和你不一样，不可能与你拥有同样的思维、习惯、性格，所以，他们不可能是你，当你要求别人要怎么样时，他们不一定可以做到，因为你的要求对他们而言可能真的很难。

我就经常犯这样的错误，比如估算工作量，我觉得 10 个小时够了，别人 10 个小时应该也没问题，就这样把任务分派给别人了，其实这是错的，他可能需要 15 个小时，或者他只要



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

2 个小时就搞定了。又如我觉得为了赶进度，少浏览一会儿网页少打几分钟游戏有时晚下班一会儿都是很自然的事儿，于是我也这样要求别人并且觉得很自然，其实这是错的，人家的做事习惯和我不一样，家里也可能有个嗷嗷待哺的小宝宝……

这都是推己及人带来的问题，冷静下来想的时候，我们也知道这样有问题，可是事到临头时却又忘记了。那我们该怎么办呢？

## 别人能教会你沟通吗

学习的本质是自我更新。从这个意义上说，学习是人的一种高级本能，并不需要教，也教不会，靠的是自觉。孔子说，“吾日三省吾身，见不贤而内省”。老子说，“自知者明，自克者强”。

学习能力是人与生俱来的，真正的学习是发自内心的，而不是别人教会你的，别人不可能教会你学习。

那么，为什么在旁观者的眼里，会认为有些人不会学习呢？我们常见的教育（学校教育、家庭教育等），它们的作用在哪里呢？

教育的作用在于发现。

每个人内心深处都隐藏着一个巨人，我们无法直接唤醒那个巨人。但我们可以当一面透亮、清澈、宁静的镜子，让别人在这面镜子中发现他们自己。

所以说，那些学东西学不好的朋友，并不是不会学习，而是还没发现自己的学习能力怎样才能爆发出来。而老师、家长、愿为人师者，他们唯一的作用就是找到一面能够唤醒这些朋友的镜子，创造一个让他们能够自我发现的学习、工作的环境，给他们机会让他们自我发现自我认知。但现实是，这样的敦厚长者太难得了，我们天天面对的是自以为是的填鸭、灌输、说教、管卡压、惩罚……

我们折回来再说沟通的事情。

上面我说了教育的问题，那么如果你想从我这篇文章里找到怎么有效地和程序员沟通的银弹，那恐怕你要失望了，因为——银弹并不存在！

现在我们来讲做一件事的过程：

- 知道做什么
- 知道怎么做
- 知道自己能不能做
- 坚持做下来

应用到沟通问题上，那就是先看自己与周围的“阿猿”们、“阿测”们、“阿品”们是否有沟通问题，如果有，是什么样的问题，这些问题怎样才能解决，自己能不能做出需要的改变来解决问题，如果能就坚持做下去。

最后我要说的是，沟通问题必须自己来解决，别人给你说个一二三四是没有用的。尽管很多问题都有章可循，但这个章却不一定适合每个人。所以，我们还是要在别人的经验里找到自己的路。

## 讲沟通的书

推荐两本书来作为结束吧：

第一本是《如何说孩子才会听怎么听孩子才肯说》。这是国外几个牛人写给爸爸妈妈的书，畅销全球很多年。可能我们的爸爸妈妈没有读过这本书，我们的很多老师们也没读过这本书，没能在我们小时候就用最好的方式和我们沟通，也没能教会我们怎样去沟通，但是从现在起，我们自己可以看着它学会怎样和孩子、和别人沟通。虽然这本书是写给爸妈的，但其实适用于每一个渴望有效与别人沟通的人。

看了这本书，可以再回头去看第二个打架故事里各位角儿说的话，你就会明白架是怎么打起来的。

第二本是《蔡康永的说话之道》，没什么枯燥的理论，都是实际的小故事，演绎他所理解的说话之道，也许你能从中找到自己的方式。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 看看你离优秀有多远

2014年8月份，我终于拿到了驾照，科目一、科目二、科目三、科目四都是一次性过的，没有给教练塞钱也没有给教练递烟。考科目三（路考）时，有个女生没能通过还被安全员讽刺，伤心地哭了，梨花带雨让人心疼。同行的两个学员考了三次都没过。我妻子的同事听说我什么致礼的事儿都没干居然一次性过了，对我表示了一些些敬仰之情。好吧，我承认这让我有点儿沾沾自喜。可是我们在谈程序员啊，和驾校有什么关系？别急，兄弟姐妹们，有的，这个真的有。

### 软件开发的未来世界

有一次我在上海 Qt 开发群( 企鹅号 279906625 )和人讨论软件开发与炒菜大妈的可比性，当时我说将来的软件开发会很简单，成为像英语、驾驶一样的基本社会能力，你能学会炒菜就能学会编程，可以根据自己的需要从半成品软件商店里购买组件，按自己的需求加工成成品软件。我相信不久之后这个就会实现，用户可以用自己的需求和智慧，借助一个平台，像搭积木一样生产软件。虽然实际上他们不懂我们现在谈论的所谓编程、开发，但确实确实，他们可以按需开发出自己的软件来，而且很可能他们都不需要懂计算机，只要在手机之类的移动设备上搞两下就好了。

其实现在硬件已经在往这个方向发展了，什么面包板、模块化之类。看看这个文章：**Make!Sense 动手好伴侣，带你轻松做实验**（<http://tech2ipo.com/88511>）。摘要如下：

“简单来说，**Make!Sense** 可以看作是一套传感器连接系统，它自带通用接口，可以快速便捷地将不同类型的传感器连接到电脑或智能手机上。作为一个有趣的学习互动平台，**Make!Sense** 最大的亮点就是操作简单，小朋友们可以用它感受身边物体细微的运动、空气中的水分、光线、温度甚至心跳，好奇什么，就感受什么。”

原文的配图如下：



## 驾校教给我们什么

驾校会教我们控制汽车，比如怎么把握方向盘，怎么使用离合、油门、刹车，怎么半联动，怎么起步，怎么半坡起步，怎么过 S 弯，当然还有调整后视镜、侧方、倒库等，对于新手来讲，这些其实也是比较伤脑筋的。但是所有这些都是相对简单的事情，大部分学员通过练习都能掌握，不会有太大的问题。多练习一下，科二、科三不成问题，至于科一、科四，做两遍题，对于没上过初中的朋友们也没太大问题，我的堂兄弟小学文化水平也拿到驾照了，去年春节回家就是他开车接的我。

当你花上五块钱买了个实习的黄牌子贴到车尾准备上路时，真正难受的经历才正式开始。这时才是真正学习开车的时刻！虽然控制汽车是成为好司机的非常非常重要的前提条件，但远远不够，很远很远。只有你看淡了风云，能够预料和避免大多数意外的情况时才能成为一个好司机。不幸的是，驾校不可能教给你这些技巧与经验。

总之一句话：拿到了驾照，实际你会不会开车，只能听天由命了。哦不，我要传递正能量，靠人不如靠己，此时此刻，只有你能对自己的生命负责，所以你必须学会应对所有的紧急状况，你必须亲自经历过很多驾驶危险并且要幸存下来才能拥有教练们企图通过说教、演示来教给你的那些应急反应能力。

舒马赫、阿隆索、汉密尔顿、阿兰琼斯、普罗斯特这些名字你可能知道。好吧，也许，韩寒这个全民岳父你更熟悉。这些知名的赛车手，还有在《头文字 D》中送豆腐却能在山道神漂移的藤原拓海，与刚走出驾校拿到驾照的学员们，到底有多大差别？到底哪些新手司机

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

可以成为未来的赛车手？

这是一个问题，很大的问题。

## 优秀是有门槛的

在“程序员零门槛吗”一节中我说软件开发门槛为零，在本文开始也 45° 角仰望星空畅想了 404 也能开发软件的悲惨未来。这里我要做一点点修正，所谓失之毫厘谬以千里，我这点微不足道的修正，会大大改变关于软件开发门槛这一话题的未来走向。

所谓的没有门槛，是指成为新手程序员没有门槛。

现在要 goto 到驾校了。

如果你想成为程序员，或者被逼迫必须成为程序员，那可以去达内、北大青鸟、黑马、野马、黄骠马等培训机构学几个月，也可以自己到软件开发 4S 店当几天学徒，基本上就可以掌握一门编程语言，能够控制某个 IDE，让它 Run 出一个 Hello World 或者别的什么。此时此刻，你满怀激情地成了 4B 青年，可是“金樽清酒斗十千，玉盘珍羞直万钱”，买个房子贷难还，娶个媳妇岳母嫌，压力不是一般的大，于是你停杯投箸不能食，拔剑四顾心茫然，最后只能辗转反侧寤寐思服……那么，你想什么呢？

终于可以连连看了，此时你就相当于刚从驾校出来的新手司机，站在钢筋水泥的丛林里，面对着人潮拥挤又车来车往的十字路口，眺望堵到霾云深处的车流，开始了事关将来的深度思考：什么时候我才能成为一个神乎其技的好司机，穿越这神鬼难破的堵局？

其实作为程序员，你也要想这个问题：我来了，面对那么多的大拿、怪咖、呆萌，我的路在何方？怎样找到那条通往优秀程序员的羊肠小道？

入门不难，优秀很难！上有六龙回日之高标，下有冲波逆折之回川。黄鹤之飞尚不得过，猿猱欲度愁攀援。怎么办？

不是每个人都能成为舒马赫，也不是每个程序员都要成为王江民。

摆在你面前的有很多选择：蹩脚的程序员、合格的程序员、优秀的程序员、卓越的程序

员，还有在职退休的好汉……

你要做哪一个？你能做哪一个？怎样成为你想做的那个？

**Big Problem !** 你需要看“自我发现，找到适合自己的职位”，step into ，或者 F11，进去看看吧。不过这篇文章目标太大，说的都是一般之理，没有针向“程序员”这个与“农民工”、“保洁员”一样响当当的称谓所代表的社会群体投去特别的温暖与关注。

## 优秀与普通的差别

有个哥们儿（叫他 GG 吧）考科三，起步、百米加减档、超车都没问题，快结束时，陪考员说，减速停车，这哥们一听暗暗吐了一口长气，心想终于通过了，然后右脚猛踩油门，陪考员反应挺快猛踩副驾的刹车，结果呢这哥们儿没有通过。

你说 GG 能否成为舒马赫？

现在我们要看看优秀程序员与普通程序员之间的差别：



## ☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

当然普通程序员和优秀程序员可能不像两幅图里的区别那么明显，但他们之间确实有一条鸿沟。

到底差别在哪里？

有以下三个方面。

- 才干
- 行为
- 结果

### 才干上的差别

请先阅读“漫谈选人与培训”了解知识、技能、才干的差别。这里我们做一个摘要：

“才干就是你那些贯穿始终的思维、感觉和行为方式，它基本上是与生俱来的。任何可以产生效益的贯穿始终的行为模式都是才干。多数时候我们并没有意识到它们的存在，只要你有意识的识别出它们并发扬光大，就可以有所成就。因为，取得出色成绩的关键在于使你的才干与工作相匹配。”

举例来讲，精确、好学、好强、严谨、秩序、乐观、开朗、专心等都是才干。

理解了才干这个说法，我是再也不迷信“勤能补拙”、“扬长补短”这些个看似励志的说法了。假如你要干好一件事情，必须要改变你的才干和个性，只有当你非你时才可以干好它，那这件事情压根儿就不适合你。换句话说，它不是你的菜。

上小学时老师总是批评有的小朋友马虎，那么马虎可以改变吗？对于有些人可以，对于有些人不可以，因为马虎是他的天性。有的人小学马虎，中学马虎，成家立业了还是粗心大意、丢三落四、马马虎虎，你让他改变什么呢？马虎就是他的天性。

还有些项目经理对团队里的程序员不满意，说 P1 要是多想想，用心一些，就不会写出臭虫乱爬的程序；说 P2 要是责任心强一些，就不会天天上班打游戏……其实，有的人就是不认真，不用心，天生就那样；而有的人就是不负责任；这些都是教不会的，改变不了的。你寄望于改变那些无法改变的东西来提高某人的工作效率和产出，那可真是痴人说梦话缘木求鱼了。

那造就优秀程序员的，是哪些才干呢？

- 学习能力
- 学习欲望
- 好奇心
- 成就欲望
- 执著
- 追求完美
- 抽象思维

## 行为上的差别

因为才干不太能直接看得出来，但它会决定你有什么样的行为。所以，从行为上，其实也可以看出普通和优秀的差别来。

优秀程序员的一些行为：

- 拿到任务，就开始仰望星空或天花板上那盏高悬的电灯棒，状如老僧入定。
- 忽而皱眉，忽而展颜一笑，忽而手舞足蹈，忽而在纸上指指戳戳，忽而口中念念有词，忽而长吁……
- 桌子上摆的是代码大全、设计模式、敏捷实践之类的书籍，并且没有灰尘。
- chrome 或 firefox 的书签栏里分门别类，类别多于 10 个，8 个以上是技术相关的
- 容不得“破窗户”，看见别人的烂代码就想改过来。
- 随手就能在白板上画出软件的流程图或者时序图。
- 项目做完了，别人在打游戏、看视频、忙着回复 QQ、向“剁手族”前进，他在想：这样重构好呢，还是那样好呢？
- 你发现他总能说出些你不知道的技术来。
- 看这厮的代码比看你自己的还好懂。
- 老得你叫他吃饭。

普通程序员的行为：

- 拿到任务就开始噼里啪啦敲键盘，一天写了好几千行代码。
- 没事儿就被测试 MM 叫过去沟通。



你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

- 都早上 10 点多了还想着昨晚的球赛，10 点半就琢磨中午要吃拉面、扯面、刀削面、牛肉面、旗花面、臊子面、窝窝面还是炒细面。
- 过了一阵子，看到自己的代码，感到很惊奇，认为是别人写的。
- 回家就看电视、打游戏、看球赛。
- 周一上班不知道要干啥，迟迟进入不了角色。
- 哇，能 Run 啦，搞定啦！
- 这谁写的代码，这么烂。算了，管它呢。
- 问别人问题的时候多，别人请教你的时候少。
- 世界上最遥远的距离不是生与死，而是你亲手制造的 Bug 就在你眼前，你却怎么都找不到它。

当然还有其他的，这个列表可以很长很长。

## 结果上的差别

一天出几次 Bug  
一天就抽你几次



最重要的差别在这里：

因为普通程序员老是和测试 MM 沟通，告别单身的几率比优秀程序员大 100 倍。

## 问题去哪儿啦

我决定花些时间梳理一下我程序员生涯中看到的各种问题，看看它们都去哪儿啦。

### 外科医生剪箭尾

在“无 Bug 不生活”一节中，我罗列了程序员对 Bug 的态度，感兴趣的同学可以跳过去看看。

程序员经常这么处理 Bug：花上点儿时间捋捋代码，却找不到 Bug 出现的根源；再花点时间看看能不能搞点沙土增高一下堤坝（想象黄河河堤的形成过程），还真找到了，我只要这里塞点代码堵一下那里扔几个判断围一下，这 Bug 貌似就不出现了。说实话，这种策略是把自己当作了外科医生，把软件当成了中箭之人。因为外科医生弄不懂内外科的分别也找不到起出箭头的办法，只好通过剪断箭尾来安抚病人，殊不知箭头还在肉里，要是烂在肉中，病人的疼痛只会更加厉害。

### 我管不着啊

现在我们开发软件通常都是兵团作战，有时我们就会发现签入到代码库中的某段代码很烂，比如各种条件分支未覆盖全面逻辑不严密，比如变量名凌乱导致可读性很差，比如类库设计不合理违反了单一职责原则……此时大部分程序员的做法是：算了，这代码是别人的，我管不了那么多，还是先忙我的吧。

有的程序员在开发过程中会感觉到软件系统设计有问题，就会认为这是系统分析师或架构师的责任，因为“我可能没他牛 X”，或者“我也没有更好的解决方法”，或者“我不大可能说服他采用我的方案”，于是就放弃了，就按“我只要实现我的模块就行了”这种方式往前走。

有的时候开发人员觉得产品某处不合理，交互方式不符合用户使用习惯，但因为这是产品经理的责任，就被“算了，我只负责实现”这种想法给俘虏了。

还有的时候阿猿觉得 UI 给的这个商品详情展示界面看着色调不协调，可还是放弃沟通

照着做出来。

类似的情况还有很多很多。

须知我们这么多人叮叮当当忙活，虽然每个人干的事儿不同，所担的责任不同，但其实是在造同一条船，没有谁的工作与别人无关，任何一处出现纰漏，都可能导致我们造出的船无法下水或者倾没深水不能远航。

## 也许问题不会在用户那里出现

很多软件在测试过程中都会出现一些难缠的 Bug：没有规律，几率极小。就连微软这样的大公司发布的操作系统，也要不断地打补丁。

作为程序员，有时遇到一个概率很小的 Bug，会因为难以重现而决定先不管它，我们会这么说服自己、测试、产品经理：这个 Bug 可能是极端操作或操作不当引起的，用户根本不会那么用，所以在用户那里很可能不会出现，就算出现了，按这种概率，也是极个别极个别的用户才会碰到，而我们还有很多其他功能没实现，不能因为这种小概率的 Bug 影响整体的进度。于是，事就这样成了。可是我们每个人头上都悬了一把达摩克利斯之剑。



要知道，用户环境比测试环境更为复杂，测试环境无法重现的、几率很小的 Bug，很可能就被某个用户碰上了，但对这个用户，发生一次就是百分百，对他来讲，根本不存在千分之几万分之几这种东西，他只知道他碰上了，你跟他说这个 Bug 实在是很难重现，真的是小概率事件，他的痛苦真真切切摆在那里，怎么可能认同你的解释。这个用户失去了，还会有下一个“幸运”用户，最终一定是星星之火燃起燎原之势，你的产品口碑会受到影响，然后你的产品十有八九就这么失败了。

我在使用手机时，像 App 闪退、App 停止运行等问题经常会遇到，给人的感觉实在是太差了，你绝对不能让我认为这是无伤大雅的小概率事件，我在情感上绝对接受不了。

## 跳过技术难题，别影响进度

软件开发的一个特点就是随时都会遇见未知的问题。比如开发互联网电视产品，很可能会遭遇音视频失步问题，具体表现是有时一个影片刚开始播放几分钟就失步，有时播放半个多小时才失步，有时超过一个小时也不失步，有时在这台机顶盒上失步，有时在那台机顶盒上失步……这个怎么办？

有一种做法是，承认这是个技术难题，一时半会儿难以解决，别让它影响进度，先跳过去干别的，以后专门抽时间解决它。

这种做法很常见，它的后果也很常见：项目进度还是无可奈何地被这个难题拖累了。

## 别人都这样

有时你会碰见这种开发团队：执行力差，产出率不高，忙闲不均。作为一个原本追求上进的你，看到这种状况往往在开始时感到不正常，想要去改变它，可是久了之后发现你一个普通程序员改变不了什么，要是你执意去做这个事儿劝那个人，反倒把关系给搞僵了，种种担心或实践让你意识到，你不是那个可以改变团队面貌的角色，怎么办？

最后的最后，现实的结果往往是：算了，睁一只眼闭一只眼吧，我管好我自己就行了。更让人不能接受的结果是，大家都这样得过且过，我也没必要这么卖力，于是自己也慢慢放松自己得过且过了。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

慢慢向生产率下限看齐，这是非常令人伤心无奈的事实。

## 我们后面会追上进度

软件项目，延期的占大多数。很多项目，干着干着就感觉要延期了。可是很多团队的项目经理、开发人员在这件事情上却表现得充满童话风格：只要解决了某个问题，只要某某猿加个班往前赶一赶，最终会赶上交付期的。

结果呢？

## 没奖金没加薪干个什么劲儿

不是每个软件公司都能持续盈利，都能大把大把地发奖金，都能隔三差五地加薪。实际情况是，相当一部分公司在艰难地生存，没有办法发可观的奖金，也没有办法按承诺的节奏加薪。在这样的公司里做开发，程序员会怎么想怎么做？

公司效益不好，与我有什么关系？不发奖金，不加薪水，我为什么还那么卖命？我何必还像以前那么努力？混混看吧。

这是一部分程序员的真实心态，我也曾经有过。要知道，当我们在一个看不到将来的环境里工作，又没有金钱刺激时，难免心灰意懒进而放纵自己随波逐流。可后来我意识到：你在公司提供的平台上通过为公司做事而修炼自己，你得到的，远不止是薪水，经验、历练、成长都是自己的，并且永远没人能从你身上拿走；时间是自己的，虚度了再也无法追回；无论何时，我们都是在为自己的现在和将来工作，而不是为公司、为老板工作。我们是一个拿薪水帮别人干事儿的人，如果停留在这种意识上，那我们一定是在扼杀自己成长的机会，浪费自己宝贵的生命。

## 还有XXX呢

受经验、能力和眼界所限，有时某个难题看起来会超出程序员的能力范围，让我们感到再花精力也是枉然，于是我们就不打算研究下去了，告诉自己：算了，还有技术大拿杨过呢，还有技术经理郭靖呢，不行还有技术副总风清扬，再不行公司会想办法，反正后面还有人……

反正我不是项目经理时有过这种想法，我当了项目经理后还是有过这种想法，甚至我做

部门经理时也曾经这样想过，就连项目总监，可能也难免有类似的想法。直到我成为一个公司的技术合伙人，我的后面，再也没有人可以接我丢下的烂摊子了。

这是多么令人不适应的情况啊！

现在我要说的是，无论何时，都应该把自己当作最后一道防线，要坚守阵地，有时一夫当关可以万夫莫开，有时一人放弃却也会导致全线崩溃失地千里，我们要想尽一切办法解决问题，这个问题到我这里就是要终止，我必须破釜沉舟不留退路。

## 反正不是我的责任

不可避免地，程序员会碰上项目延期这种事儿。有的人会回顾项目执行过程中的种种问题，包括反省自己的不足、梳理团队成员协作上的问题、琢磨任务安排与进度跟踪是不是出了差错……这是积极的做法，我觉得每个程序员都该这么做，然而实际情况却并非如此，有相当一部分人会这么想：领导安排的事儿，我该做的都做了，我有什么办法，这是大家伙的事儿，很可能是项目没管理好或者需求老变来变去，跟我没有关系。

到底什么时候能完成？



## 算了，换个环境

程序员换工作的频率很高，有的人两年换一次，还有的人一年换一次，更有的人一年换几次，这么频繁地换工作，真实的原因是什么呢？

有时是干得不顺心，觉得自己空有一揽子想法奈何领导不重视；有时是觉得团队的氛围不行，不能让大家安心做事，自己也无法保持积极向上的心态，在技术和职位两方面都无法获得成长；有时是觉得产品没有希望，看不到未来，越干越迷惘；有时是别的公司的超人气美女抛来了酸爽无限的绣球；有时就是莫名地，觉得该挪挪地方了……

换个环境吧，也许会好一些。果真如此吗？

你最终决定跳槽，一定是发生了什么问题，导致你无法容忍现状，但一定要问自己：我现在面临的问题，究竟是公司、团队、产品的问题，还是自己的问题，换个环境是不是不会再出现类似的问题？

也就是说，我们决定跳槽之际，需要深入分析，问问自己的内心，到底是外在的原因导致自己再也无法忍受现状，还是发自内心地觉得自己需要做出改变，去寻找新的方向。我们要弄明白自己的期望，搞清楚自己想要什么，这样才不至于刚出泥沼又入火坑。

作为程序员，在开发和工作的过程中，难免会遇到各式各样的问题。面对问题，每个人都有自己的处理方式，**不管你采取哪种方式，问题都不会自动消失**。假如你采取的措施暗合“希望问题自行消失”这种模式，一定要注意，这是非常低级非常有破坏性的习惯。用这种模式“解决”问题的次数多了，时间久了，我们自己慢慢就成为问题了：因为我们事实上是在逃避问题，拒绝成长与成熟。我们希望跳到一个问题很少的环境里开开心心地工作，可这样的环境无异于理想国和桃花源，它们只存在于想象之中，如果我们自己的内在没有发生改变，不能培养出接纳问题、解决问题的能力，将在寻找的路上疲于奔命却永远也找不到这样的地方，最终绝望，最终选择“放弃自己”这一看似最不可能的结果。

既然各种各样的问题不可避免，那唯一可行的就是：直面问题，解决问题。

软件开发问题多多苦难重重，只有接受了这个前提，只有理解了这一生活与工作的本质，我们的工作与生活才会变得美好起来。

我们往往在问题面前缺乏耐心，想让问题马上解决，如果花费一部分时间和精力还解决不了，就想丢在那里不管，迈步跨过去或绕道而走，我们总想尽快脱身，尽快缩短与问题接触的时间而不愿花足够的时间来应对这种不舒服的感觉，不愿冷静地分析问题。因为直面问题解决问题，真的是需要绝大的勇气和卓越的耐受力来承担切肤裂心的痛苦。

然而，我们必须面对。如果选择逃避，只会越来越糟，问题会跟着我们走，我们走到哪里它们就撵到哪里。套用香港警匪片里的经典台词：“出来混，迟早要还的”。我们也一样，遇见问题，逃是逃不掉的，越逃问题越多，到后来总有你要还的那一刻。

既然如此，何妨早一日受苦、早一日解决、早一日浴火重生？让我们接受“先苦后甜”的生活模式，坦然面对问题。问题可以开启我们的智慧，激发我们的勇气。为解决问题而努力，我们的思想和心灵就会不断成长，心智就会不断成熟。我们解决问题的能力就会不断获得提升。我们终将因为直面问题而淬炼成钢，在混乱中成就自己，最终成为独特的自己，找到自己的方式活出个样子，这就是成功了。



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 怎样成为技术达人

程序员身边总是有优秀的前辈、后辈或者同辈。优秀的人总是让人羡慕。比如自己苦搞一天的 Bug，头发快抓得掉完了，人家扫上两眼，改一行代码，问题就解决了；比如自己干了十年开发，一个月薪水还不到 10K，人家才毕业就签了 16K 月薪的合同；比如我要天天辛苦加班工作还干不完，人家什么事儿不干每天就喝喝咖啡，结果写出来的软件居然比我的稳定多了好用多了。这样的事情太多太多，怎么办？

陈胜起义时说：王侯将相宁有种乎！

作为程序员，面对优秀的同行，其实我们也可以愤而追之：大丈夫当如是也！

有梦想有追求总是好的，关键是，如何才能更优秀呢？

### 程序员的分类

早上翻看陆其明老师翻译的 Jeff Atwood 的书《高效能程序员的修炼》，里面有一篇文章，名字是《程序员的八种境界》，讲得很好，受益匪浅。不过我打算从另外一个角度为程序员分一分类。

### 10000 小时法则

迈阿密大学的 Anders Ericsson 教授写过一篇论文——《有目的的训练在专业人才培养中的作用》(The Role of Deliberate Practice in the Acquisition of Expert Performance)，介绍了他在研究“最好的音乐家”有什么共同点时的发现：卓越的音乐家的练琴时间，没有一个人小于 10000 小时。

2008 年，畅销书作家 Malcolm Gladwell，将这篇论文写进了他的新书《异类：成功人士的故事》(Outliers: The Story of Success)。

Malcolm Gladwell 概念化了原始论文的结论，宣称存在一个“10000 小时法则”(10000 hours rule)，即成功至少需要 10000 个小时的投入。

这本书出版后，成为《纽约时报》畅销书排行榜的第一名。

10000 小时法则背后的思想是：**真正决定一个人成就的，不是天分，也不是运气，而是严格的自律和高强度的付出。**

我想很多人看过古龙大师的《边城浪子》，里面的主角是傅红雪，资质平庸又身有残疾，结果却练成了举世无双的刀法，其秘诀非常之简单，卖油翁一言概之：无他，唯手熟尔。要知道，傅红雪被不是妈妈的妈妈每天逼着练一本残缺到只有拔刀式的白家刀法，苦逼地每天拔刀两千次以上，拔了十几年，你试试，要是你拔刀拔十几年，不疯掉的话肯定也能炼成个无影刀什么的。

傅红雪是另一个 10000 小时法则的实例。

## 爱是众妙之门

我在前面介绍了“10000 小时法则”，当然那只是一个必要条件，不是充分条件。我以为要想让 10000 小时法则有效，还有一个条件。

安·兰德说过：“为了将事情做好，首先你得喜欢做这件事，而不是喜欢这件事情的结果，那仅仅是第二位。”

想成为大牛程序员，不能仅仅盯着一个项目干完后的奖金，也不能老想着忙过这阵子就可以休息了吧，也不能盼着应用能 Run 就交差了，也不能老想着我学了 Scala 就能拿到 20K 的薪水……

大牛程序员很多是这样的：喜欢干，干得牛出成绩了，收益来了，机会多了。

要说你不喜欢一件事情，有时被逼着也能胜任，但要说你讨厌一件事又能干得出类拔萃，那我不相信，最好你也别让我相信——我怀疑你早晚有一天会人格分裂。

伟大的科学家爱因斯坦（他的宗教观也很有意思 [http://en.wikipedia.org/wiki/Religious\\_views\\_of\\_Albert\\_Einstein](http://en.wikipedia.org/wiki/Religious_views_of_Albert_Einstein)）说过：“兴趣是最好的老师。”这就是说，一个人一旦对某事物有了浓厚的兴趣，就会主动去求知、去探索、去实践，并在求知、探索、实践中产生愉快的情绪和体验，而这种愉快的情绪和体验反过来又会促使他继续去求知、探索、实践。这样就产生了良性循环。“成为达人”的欲求和“痛并快乐着”的体验就会推动着你不断前行。

你爱编程，享受做设计、写代码、解决问题的过程，再遵循 10000 小时法则，那你想不牛都难。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 伟大的方法论

前面是开胃酸汤，现在到正题了，我们来聊聊一个程序员，要如何才能一步一步慢慢牛起来。

### 编码



毫无疑问，你必须得编码，持续不断地编码，要达到一定的量。

量变会引起质变，但不是必然。

持续编码的结果有两个，一个是达到质变，水平不断提升；一个是重复重复再重复最终麻木。

要哪个？

### 回顾与思考

要想质变与提升，必须有目的地停下编码！

只编码，不走心，那是万万不行的。不走心的程序员很多，干多少年了水平还和初干时

差不多，十年如一日。走心就不一样，走心一日如十年。

我们要周期性地回顾自己，看看以前的代码，分析一下现在的设计，看看哪些可以改善，有多少种改善思路，这样就能看出以前的自己与现在的自己这两者之间的差距。如果你觉得自己以前的代码让你看不下去，那恭喜你，你绝对提升了不少。

要回顾以前，还要分析现在。程序员还要经常梳理一下自己的状态，看看我现在都掌握了什么技能，在哪些方向和领域有所成就，再加上对过去的回顾，看看哪些是延续性的前进，哪些是无心插柳，哪些是冤枉路，哪些是意外之喜。这样就会对自己的能力边界有所了解，知道自己知道什么，也知道自己不知道什么，更重要的是，你会发现自己能把什么东西学好用好，还能发现自己为什么可以把这些东西学好，那样你就可以找到自己擅长的方向，做得更好。

回顾了以前，分析了现在，还要想想将来。在技术上，我要在哪些方向继续前行深入挖掘？这一点非常重要。我们伫立于现在，回首过去，仰望未来，值得期待的只有前方。

有人说上到山顶就只有下山了，这也是真的，不过，顾城有首诗《避免》可以看看：

“你说

你不爱种花

因为害怕看见花一片片的凋落

所以

为了避免一切的结束

你拒绝了所有的开始”

作为有追求的程序员，我们不必这样担心，只管积累我们的 10000 小时，只管思考如何有效地利用这 10000 小时，不单过程是美丽的，将来也会是美丽的。

## 多学一点点

台上一分钟，台下十年功。那些优秀的程序员，无一不是付出了比普通程序员更多的精力。

我们在开发时，会接触各种各样的新技术。有时因为项目时间原因，很多技术只能是走

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

马观花地用，以解决问题为目的，项目结束了可能就扔一边儿去了。但要想让自己有更大的成长，我们就应该多学一点，用一分学两分，更广一分，更深一分，不能不求甚解一带而过。对于用过的技术，最好在当时，项目的间隙或者项目结束时，抽时间了解它的背景、长处、机理，让自己对用到的技术都有相当程度的了解。惟其如此，我们的知识面和技术深度，才会随着开发经验而不断递进。

## 实践新技术

选择或者学习一门新的技术，必须要实践。有机会的话一定要把新技术应用到实际项目中，如果工作条件限制了，那也要自己定义一个自己喜欢的项目，坚持使用你的新技术做下来。唯有实践，你对新技术的理解才能更为深刻。

知道有那么一门技术，与实际应用过那门技术，其间的差异可不止一点。

## 沟通



很多人觉得程序员比较难沟通，说程序员不会好好说话，你觉得呢？

我想说的是：要想成为优秀的程序员，沟通能力一定要过硬！

程序员是干嘛的？不是写代码的，而是解决问题的。

### **解决问题才是程序员的职责。**

要解决问题，离不开沟通。

客户会找你，市场会找你，产品经理会找你，UI 设计师会找你，开发同事会找你，测试 MM 会找你，老板也会找你，女朋友会因为你的投入与执着爱上你，也会在爱上你后因为这些而不断与你争执……

哪一样离得开沟通呢？

当然我不是说程序员要像销售、市场、营销等岗位的兄弟姐妹们一样妙语生花，其实沟通最重要的是能听懂别人说什么，能把你想说的以对方能理解的方式说出来。至于是妙语连珠还是磕磕绊绊，是语言沟通、眼神沟通还是书面沟通，只要达到目的即可。

有人说自己天生不爱说话，跟机器打交道才觉得踏实。其实这没关系，你要找到自己的沟通方式，看怎样才能让别人理解，也调整自己的聆听和理解方式，看怎样才能理解别人说的话。

怎样才能让别人正确理解你的话，怎样你能正确听懂别人的话，这都需要不断的学习、练习，这方面有很多的书和视频，可供参考。这没什么不好意思的，我认识一些朋友，觉得学习说话技巧是很不好意思的一件事儿，好像这样做就说明自己心怀鬼胎就多丢人似的。其实不必，我们的目的不是巧舌如簧地骗取别人的信任然后来谋取利益，我们的目的很简单，就是让别人理解我们也让自己理解别人。

## **理解沟通模型**

沟通这事儿真是一门学问，虽然我在这方面一向鲁钝，还是想以我们程序员的方式，打个比方来说明一下沟通模型。

首先每个人心里都有一个 TCP Server，监听到某个端口上，等待陌生人或者熟人或者半生不熟的人来访问。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

其次呢，不同的人对自己的端口是否开放持不同的态度，OPEN 的人就会监听到知名端口，比如 80、21、7、13、22、23 等，这样别人就知道我们找这个 OPEN 的家伙说事儿该连接哪个端口。当然，只开放端口还不够，灵慧的人还会慢慢把自己的协议 SHOW 给别人，这样想找他聊天的人就可以遵循这个协议来说，事情就容易多了，双方就都能理解。

还有一些人呢，比较内敛，虽然内心很渴望别人找自己聊聊天谈谈心，可是不知道怎么把自己的端口开放给别人，或者出于某种原因隐藏了端口。比如他本来是监听到 21 端口上，走 FTP 协议，因为被别人通过端口扫描的方式黑了两次受了点不大不小的伤害，就小心地封掉了 21 端口，把自己的服务挪到了其他端口上，并且也不告诉别人。这就出问题了，别人找不到端口号，怎么和你沟通呢？

还有一些人，自己也不知道该监听到哪个端口上，不能有意识地训练自己，今天是 21，明天就 80，后天又走 37 了，结果不但自己混乱，跟他打交道的兄弟姐妹们也混乱：昨天这么说还挺融洽，今天再这么说他居然翻脸了呢！

这是我们每个人内心的 TCP Server。其实每个人还有一个 UDP Server，是为那些夹生的人或者拥有特别标签的人准备的。你可以知道我的 UDP 端口，但是你发消息过来我不一定搭理你，被拆穿了我还可以说 UDP 协议不可靠，我没收到……这两个 Server 就是我们每个人身上基本的沟通模型。

最后，如果你要想向别人说明你的意图并让别人按你的意图去做某件事，那就要花费精力去研究如何找到对方监听的端口、使用的协议。和 OPEN 的人沟通比较容易，访问 80 端口，走 HTTP 就行了，或者夜半三更翻墙过去幽会也能成。而针对那些隐藏了端口和协议的，就要启用端口扫描这种神器，扫到端口，再抓包分析协议，成功的可能性总是有的，只不过费些力气罢了。

用这种模型来解释现实生活中的沟通问题，程序员们就可以释然了：啊哈，原来我跟他说的的事儿他老不反应，是搞错了端口或者走错了协议。

## 表达自己

理解了沟通模型，找到了对方的端口和协议，还有一个问题需要解决：你怎么准确地表达你的意图。

对于相当一部分程序员而言，这是个难题。有的人本身就寡言少语（我是这类的）；有

的人很厉害，与别人说话时脑子转得比别人快几倍，脑子里盘旋的是长长的带上下文的句子，从嘴里出来后就成了嘎嘣脆的孤零零的词语，还以为自己说过那些上下文了别人应该都能理解这些不着四六的短句；有的人写代码一二三四条理清楚，说话时就比较发散，正说股票呢忽然就转到 APEC 蓝上了；有的人比较羞涩，大男人说话却声若蚊蝇……总之各类情况都有，你见到的绝对比想到的多。

不管你属于哪一类，要想顺畅地与人沟通，先要训练自己，让自己可以准确地表达。

我是不会表达的人，你看我洋洋洒洒长文千字，其实我不大会说话。不过我要说的是：如果你说不好，那就想办法先写好。

“写”这个事情，比说更容易让人条理清楚，因为写这个过程时间相对宽松，上下文简单，不像与人沟通那样你来我往还要时刻注意话外音，整个过程比较复杂。所以，写，是一个很好的锻炼自我表达的手段。写作能训练你表达的准确度。

你可以通过写博客、写日志等手段，把自己的想法表达出来。然后过一段时间，你能通过写作来表达自己了，再回头来影响说话，你把嘴上说的当作纸上写的，多练习练习，就能准确地表达自己的意图了。

## 找到你的小伙伴





☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

程序员干的是技术活，找到一个谈得来的小伙伴很重要，可以相互促进，消除登顶路上的寂寞。

可能你需要在不同的阶段找不同的伙伴，最好是找某方面比你牛的人一起研究技术，这样可以学到东西。

或曰：牛人为什么要搭理我呢？

“在你所有的能耐中，至少应该有一样是拿得出手的。不会做很多事情并不可怕，可怕的是没有一件事情能做好。”

要培养自己在某方面的长处，这样就可以互补了。

其实有相当一部分程序员离职的一个原因是：在这个公司真的太寂寞了，都找不到一个可以讨论技术相互促进的人。

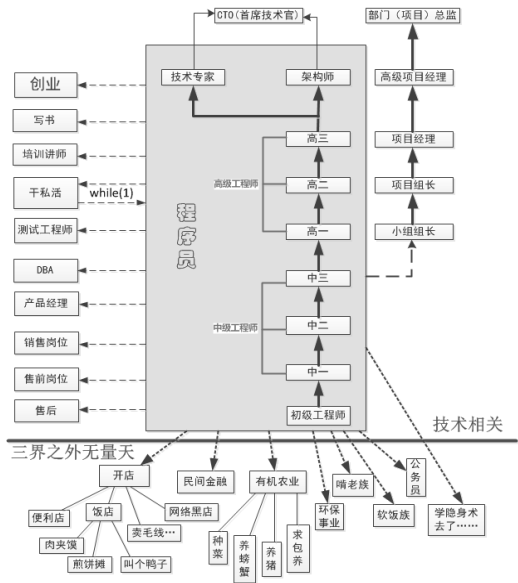
## 分享

分享，真的很重要，千万别掖着藏着，对待向我们求助的兄弟姐妹，要像春天般的温暖。

此处略去 5 万字……

# 一张图道尽程序员的出路

最近在琢磨程序员到底路在何方，经过不断的“自虐”和代入，终于在“迷雾森林”中得图一张，看之豁然开朗。独乐乐不如众乐乐，分享给大家：



程序员出路图解

## 术语表

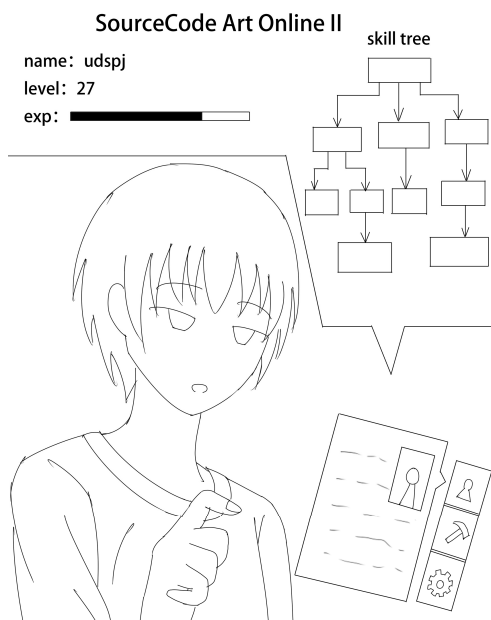
图中使用了很多术语，这里先做解释，只有我们大家都说同一种语言，沟通才会顺畅。

- 程序员：撰写代码，生产软件，辅助提高社会生产率的脑力劳动者。据说喝的是咖啡，抽的是烟，产的是代码。他们是计算机诞生后才逐渐兴起的一个群体，鱼龙混杂，有高飞天上的“白富美”，也有低到尘埃里的“矮矬穷”，总之虽一类而九流具足。
- 初级工程师：入门程序员，怀揣着 IT 行业多金的遐想进入了 IT 江湖，开始了练级打怪求升级的游戏之路。

## ☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

- 中级工程师：经过几年的打拼，有一定的拷贝、粘贴功底，掌握了太祖长拳或罗汉掌之类的一技之长，打字速度变快了，双眼变得稍稍有些迷人了，对零壹世界有了比初级工程师更深刻的认识。
- 高级工程师：拷贝、粘贴之术出神入化，又习得搭积木和沙滩上盖大楼的绝技，还能像江湖郎中一样给病入膏肓的代码开药。
- 架构师：坚信技术可以让人更优秀，抵制了各种旁门左道的诱惑，闭关多年，终于一生二、二生三、三生万物了，能够坐而论道、指手画脚了，觉得自己形神具备有仙人之资，偶尔来个仙人指路挺爽。
- 技术专家：看过各种江湖流派，最终在某条羊肠小道之上月黑风高之夜顿悟辟邪剑法，从而成为一代旁门高手，在自己的领域里无人能及。
- CTO：史上最强的技术流？不一定。不过有一点高度是肯定的：会当凌绝顶，一览众山小。正统的 CTO 专门研究各种武技，寻求能在未来的江湖中克敌制胜的秘辛。国内江湖乱象纷呈，很多 CTO 其实在打杂做管理。
- 小组组长：这并不是一个正式的头衔，在《江湖异闻录》中，小组组长被掌门委以突袭带头人的角色，只因为他有带头大哥的潜力，能激发大家的战斗力。虽然没有掌握绩效考评之类的绝杀技，不过也是通往管理之路的必经过程。
- 项目组长：管的事儿有点儿多了，要带人，要带项目，要和产品经理搞好关系，通常也得和小分队的成员不分你我，总之什么都干，干的是项目经理的活，只是没有名分。没关系，等带好一个项目，又有项目经理的缺时就可以顶上去了。
- 项目经理：这是一个正式的打杂的，终于有了名分了。
- 高级项目经理：又叫项目群经理。他搬了把凳子以便让自己站得更高，能够看清楚纷乱的江湖里多个项目的情况，通常需要协调这个协调那个，找项目经理谈谈话，抚慰一下项目经理或项目组长委屈的心，有时也会被来自上面的压力压得喘不过气来。
- 部门（项目）总监：不知道要说什么了，开始管各种事儿了，评估项目开发过程，拟定考评、绩效、职级等各种制度，或者今天跑这个项目聊聊天，明天跑那个项目聊聊天，或者自己左右互搏和自己聊聊天，看起来闲人一个，实际上骚年的心从未停歇。

就这么多吧，其他的也不想说了。因为我们要讨论的是程序员的出路。



## 程序员的路在何方

前面那个图中框起来的，是程序员的主旋律。

程序员，专业人士也，与瓦工、木匠、发型师、保洁员等类似，不过是社会万千分工之一，没什么“高富帅”，也没什么“矮矬穷”，只不过是时代大潮中被滚滚洪流裹挟着前进的一群人而已，所谓泯然众人矣，就是说的这个。

## 你怎么混进来的

不忘初心。

不忘初心。

你是不是忘了当初为什么要干这个？

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

那就想想吧。有好处。

有的人是为了程序员的高薪；

有的人是想要一份体面稳定的工作而已，别出差；

有的人是过来“打酱油”的，早抱定了过两年就走的心；

有的人想了解这个世界，试试看自己合不合适；

有的人就喜欢计算机的世界，如鱼得水；

有的人觉得这个行当既专业又神秘，感觉倍儿棒；

有的人喜欢挑战，因为软件开发每一刻都有战斗的感觉，各种问题纷至沓来；

……

总之各种理由都有，你也一定有自己的那一个。

## 将来去哪儿

前几天在朋友家翻看米兰昆德拉的《不能承受的生命之轻》，里面开篇对轻与重的辩论，很有意思，好像和我们也有点关系，感兴趣的可以读读，这本书是倒退十来年那个时候的小资必读。

你今天的选择，决定了明天的路。

那些来“打酱油”的，明天就走了。比如我原来有个同事，干了一阵子，来找我辞职，说准备回家搞民间金融。后来还在 QQ 上联系过我，据说搞得还不错，问我要不要投资。

有些人听说 IT 行业薪资高，眼巴巴地进来了，可是干了两三年，工资还是三两千，只好失望地走了。其实除非你有经天纬地之才，否则没有一开始就高薪的道理。先做事，后有 钱。先想钱，钱难来。做程序员搞开发大抵是这样的。

听说有个哥们儿跳到了华为，天天晚上工作十一二点，有一天他终于不慎抱恙，晚上九点多回去了，被整个组的人鄙视；在华为熬过三年，收入很多，年终奖和分红抵得上你一年的工资。关键是，你熬得住吗？

我想说的是，作为程序员，心态决定你的将来。你自己怎么看待你所从事的工作，你是享受它、厌烦它、无所谓、爱恨交加……这些决定了你将在这条路上走多远。

我们生活在一张社会关系网中，因为别人能够看到你，你才存在，这是“作为意识与表象的世界”，你看到的人、别人看到的你，都只是作为观察者的那个人想看到的，而非真实的人。虽然真实的人客观存在，但没有人能看到这样一个真实的、完整的人，包括你自己。

你看到你想看到的，他看到他想看到的，张三看到张三想看到的，李四看到李四想看到的……周围一圈人对你的意识构造了社会化关系中的你，但那只是你的一小部分，仅此而已。如果别人不知道你，你就不存在。为了存在，你就要在别人面前出现，这就是“存在感”。

所以现在很多人在刷微信、微博、博客、论坛、陌陌，都是在刷存在感。当然也包括我自己。

让别人知道你，你才存在。你只有留下点让别人难以忘怀的东西，你才因这些东西而存在。

如果想明白了这个道理，一个真正热爱技术的程序员，想在这个行当里干点儿事情的兄弟姐妹，就不会去在意别人说什么“软件开发是青春饭”、“程序员平均寿命低于普通人”、“程序员群体社会地位低下不如鸡”、“过了 30 岁就要另谋出路”之类的话。作为一个与其他行业没有什么差别的行业，仅仅是分工不同而已，为什么有这么多说法呢？众说纷纭，都是杂音。有用的话不这样，大音希声。

## 程序员的两条主要通道

看前面那个图就很明白了，程序员有两条主要的职业通道：技术和管理。

在中国有个很不好的传统：学而优则仕。

如果你在一个单位干技术干了很多年，还当不上领导，就会被人瞧不起。这也是很多人干开发干了几年后，正当年富力强生产力旺盛的时候脱离技术通道的原因。因为领导大部分都是这么一个套路：“干得好？行，带人吧。带得好？行，升经理吧……”所以，很多原本可以成为技术大牛的人，就这么被拐走了。

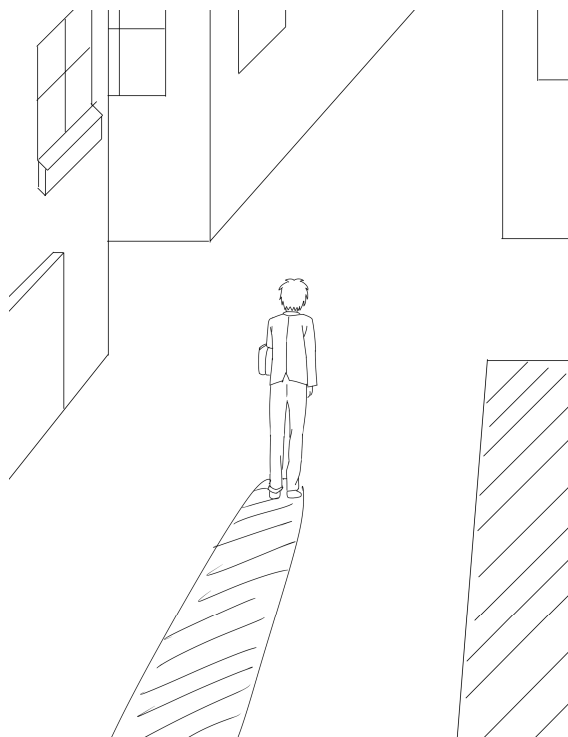
也有一些专注于搞技术的开发人员，就要走技术通道。

你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

有个哥们儿，做 CMS，前台后台都能做，领导让他带三个人，让他当项目经理，他不乐意，说我自己干就行了，管别人净浪费时间。领导好说歹说把他绑在了项目经理的位置上，结果他不管那几个小伙伴，也不安排工作给人家，人家来问他这个怎么干，他就说你们不用管了，我来干。最后呢，一个人干了 80% 以上的活儿，其他几个人闲得头顶生了蛛网。再后来呢，领导不住地找这哥们儿谈话，想教他怎么做项目经理。再后来这哥们儿就离职了。

“学而优则仕”还有一个非常不好的地方：普遍认为当领导的薪水要比搞技术的高。这也是很多本来有灵气的开发人员想走管理路线的原因。我想干技术啊，可你们看不起我，不给我那么多钱，周围人也这么看，我没法老这么干下去。

其实如果你不在意这种周围的想法，做技术未尝不是一条幸福的路。



## 跳出三界外

前面我提到有同事搞金融去了，还有别的故事。

我原来有个同事，写程序很厉害，是公司的高级工程师，后来不干了，开便利店去了。

前几天有个朋友说他找了个合伙人包了点儿地在搞有机农业，种菜。

有个阿媛在淘宝上卖内衣，后来把工作辞了。

有个阿猿开了家咖啡馆，名字就叫程序员咖啡。

还有的阿猿专职做 Uber 司机了。

这都是跳出三界外的故事。其实也很平常，你的选择，你做主。如果你觉得这个行当不是人待的地方，再也不要受这罪了，那就走吧。如果一份工作带给你的痛苦比欢乐多很多，确实没有留恋的必要。真的，你肯定是走错了路。

我在看《高效程序员的 45 个习惯》时，发现第一章里引用了一句土耳其谚语，写在下面，送给诸位，也送给我自己。

**不管路走了多远，错了就要重新返回。**



## 技术人员为什么要写博客

很多年前，我曾在榕树下、西陆社区、天涯社区、红袖、网易等文学站点写小说，那时也有人问我，写这个做什么？现在不写了，只偶尔在心中构思、琢磨，但我想某年某月某日的某个午后，也许我还会拾起文学的笔来。

我妻子曾经在我晚上写技术博文时问我，写这个做什么？当时怎么回答她已经忘了，但今天我又想来说说个中缘由。

我为什么写博客呢？

### 首先，主观为自己

写博客对自己的提升，毋庸置疑，是很大的，我想写过博客的人都有体会，大概有这么几点。

### 强化经验和知识点

为了写一篇文章，你会对代表文章中心的知识点以及它所涉及、衍伸的知识点进行了解、学习，甚至以前与此相关的记忆也会被唤醒，被重新组织，经验会沉淀下来，你所拥有的可能原本散落在草丛中的知识点和经验都会在此时此刻被发现、关联、系统化。我们的知识点可能是散落的珠子，而写博客则充当了衔接它们的丝线，一旦经验被有效组织，那么就得到了再次强化。学而时习之，不亦乐乎。

### 可以索引的记忆宝库

有时我们会忘掉以前的东西。其实我们的脑子就是记忆体，旧东西可能会被新东西覆盖，生活重点、工作重点的转移，都会让你淡忘一些、送别一些而又迎来一些新的技能、知识、经验。

多少年以后，我们很难想起当年那个在蹒跚学步的自己遇到的那些事儿。而博客，作为一个载体，既可以满足我们的表达欲望，又可以供我们以后凭吊当年。

## 提升学习能力

每个人都有自己的学习方法，这一点在上学时就千差万别，在工作中更是如此。

当你接手一个新任务时，你会怎么做？

当你要用一个新技术时，你会怎么做？

每个人都不一样。而在学校学习、在单位工作的过程中，我们会发现、识别自己的学习模式，有些人是下意识的，有些人是有意识的。而我们在写博客的时候，在回顾总结的时候，会把下意识、潜意识变成有意识，写博客的过程，有助于我们回想自己的学习过程，强化我们对自己的学习模式的认知，最终能够清晰地表述自己的学习模式，而识别到了自己的学习模式，潜意识变成了有意识，那么在后续接触新技术新知识时，就会更快。

## 锻炼逻辑思维

万物皆有关联，某个特定的技术和某个特定的人一样，都处在一张网中，是一个结点，牵一发而动全身。那么这个点和其他的点，是什么关系？为什么是这样的关系？

我的女儿被老师戏称为“十万个为什么”，在理清这些“为什么”的过程中，我们的逻辑思维能力也会得到提升。

## 锻炼文字表达能力

为了写好一篇文章，你需要遣词造句，不断琢磨。很多人都写过 PPT，可是有多少人写得到位呢？

为了让自己的博文能够流畅，有吸引力，你会选择一个好的组织结构，包括段落的起承

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

转合以及词汇的准确传神，都是需要考虑的。在写博客的过程中，在坚持的过程中，你会发现，自己越来越能写、会写了。

## 其次，客观为他人

有句话，赠人玫瑰，手有余香。

我们在做项目的过程中可能会遇到问题，而张三遇到的问题，李四也可能遇到，还有王五赵六。当张三把他解决问题的经验记录下来、分享出来时，后续的李四等人就可以少走弯路。这是无意识的共享。

好为人师，这是很多人的共性。有时我们希望帮助别人，用自己的经验帮助别人少走弯路。

## 博客给了我什么

那么，写博客对我来讲，究竟有什么收获呢？

我从 2013 年下半年恢复技术博客写作，坚持至今，得到了什么呢？

- 锻炼了我的自控力，让我发现自己还是有一些事可以坚持做下去。
- 我因为博客，与出版社结缘，出了《Qt on Android 核心编程》《Qt Quick 核心编程》两本书，以及本书。
- 我还录制了《Qt 基础和 Qt on Android 入门》《JavaScript for Qt Quick(QML)》两门视频课程，发布到了 CSDN。
- 认识了一些热爱技术的同道中人。
- 2014 CSDN 博文大赛编程组冠军。
- 获评 2014 CSDN 十大博客之星。
- 我的初中校友因此找到了我，把我拉回了集体中。

路还在走，未完待续……

## 程序员和女朋友（男版）

什么？没有女朋友？那也没关系，先看看这一节有助于你早日脱单。

想当然



这家伙肯定还没出发！

会议还没结束……

讲个小故事。

我爱人在一家公司上班，我俩单位距离不太远，有一天快下班的时候领导找她讨论 PPT，估计赶不上班车了，就打招呼让我骑着单车去接她。

妻说：“领导找我讨论 PPT，还不知道几点结束呢，你来接我一下。”

我说：“我现在过去。”

☎你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

妻说：“你过半个小时再过来吧。”

过了十分钟，我骑上单车出发了，10分钟抵达她们公司楼下。抽烟，在手机上看张小娴的《禁果之味》。

转眼七点了，她还没下来。我有点儿着急啊，那虽然是一个晴朗的冬日，可天早黑了，风飕飕地割在脸上，昏黄的路灯也不能给人丝毫暖意。我又不好意思打电话给她，于是我就瑟缩着身子继续看《禁果之味》。

七点十五分，我爱人电话来了：“你在哪儿呢？”

我一听这口气不对啊，我就说我在你们楼下等半个多小时了。

妻说：“怎么不给我打电话，我早开完会了！”

我说：“怕影响你开会，以为你开完会肯定要给我打电话。”

妻说：“你是不是还没到？肯定出发晚了。”

我说：“你赶紧下来吧，冻死了。”

妻说：“你为什么不给我打电话？”

……

我理解错了吗？

我爱人每天送女儿上幼儿园，女儿都叮嘱：“早点下班啊，一定早点下班啊。”

这不，女儿还在家等着呢，可是你看我们居然互等了半个小时还多，这是典型的锁等待啊。

一路上我和媳妇就“我到了该立即给她打电话”还是“她开完会了应该立即给我打电话”这两个问题“和谐”地讨论了一路，谁也没能说服谁。

过了三四天，有一次聊天，媳妇说：“我问了公司好几个程序员，他们反应居然和你一样！”

哦，是么，看来我们都比较容易接受回调和订阅模式。

好吧，这只是我生活中的一个小插曲。我从这个插曲里收获了一点点小心得：要是你遇见类似的问题，直接承认“是我应该先怎么怎么着”就 OK 了。这样可以避免争执，男人的胸怀都是委屈撑大的。再说啦，在这种事情上和你那位较真儿没什么意义，还是把精力留着折腾代码吧。

## 你为什么跟我没话说

程序员一天到晚和电脑厮守，很多人关注的也都是 IT 江湖的那些事儿，不大爱瞎聊。如果别人提起一个话题，感兴趣了可以海阔天空唾沫横飞甚至面红耳赤激情洋溢地扯上好一阵儿，不感兴趣了就是“嗯”、“哦”、“对”、“是吧”之类的简单回应。女朋友提起的话题多数是后者。

比如女朋友在家里试衣服，左挑挑右挑挑，对着镜子左顾右盼半天，远远地问你（你在电脑前写代码或者看程序员的段子），“阿猿，这件好看吗？”，你眼睛都没离开屏幕，“好看”就脱口而出。女朋友这就恼了，“你看了吗就说好看，一点儿诚意都没有”。

比如女朋友和你说起单位谁谁谁今天穿了件裙子特别漂亮，你通常会“嗯”一声。

比如女朋友说“好烦，今天又瞎忙了一天，累死了”，你通常会说“嗯”。

比如女朋友说“今天单位 XXX 居然说我上班看微信太频繁影响工作，太气人了，她怎么能这么说呢”，你可能会回应一句“是吧”。

比如妻子对镜贴花黄时可能会来一句，“哎呀，有鱼尾纹了”，你可能会说，“是吧，到年纪了”。

比如妻子会在三八妇女节时对你说，“今天是三八妇女节呀”，你回她一句，“是啊”。

……

有没有类似的情况呢？

你下班了，会在家里继续与电脑厮守，有时学点儿新技术，有时开发你自己的 App，有时游戏，有时与别人在 QQ 上聊聊天，有时脸上挂着蒙娜丽莎的微笑，有时自个儿“嘿嘿”

## ☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

直乐，有时面色凝重眉头紧锁，有时自言自语“不可能呀”、“怎么会呢”，而你的另一半，则陷在沙发里拿着遥控器研究小米盒子里哪个连续剧好看，一会儿问你“这个好看吗”，一会儿问你“那个好看吗”，你听得烦了，要么头也不抬地说“好看”，要么敷衍一句“你自己选一个你喜欢的就行了”，要么干脆不吱声。女朋友不乐意啊，走过来粗暴地把你的笔记本合上，质问你“你为什么跟我没话说”，这时候你虽然小有点儿不满，还是压抑住脾气，自以为好声好气地回了一句，“哪儿有啊，这不一直和你说话呢”。当你女朋友掰着指头追问“半个小时了，你一共说了八个字”时，你有点不耐烦了，回了一句“别烦了，我忙正事儿呢，程序出了个 Bug，怎么找都找不到”，于是……

……

我有时候是这样子的，我的一些程序员朋友也有类似的经历，其实我们程序员也很愿意说话，关键要看和谁说、说什么。每个人都乐意就自己感兴趣的话题聊一聊，天生闷葫芦不在讨论范围内，哑巴也要排除在外，还有植物人。这是程序员的说话与写作方式，老是限定一堆条件，看（听）着不像人话是吧。

当女朋友问你“你为什么跟我没话说”时，你怎么办？

否定她，“哪儿有，我一直在和你说话啊。”

哄她，“我最喜欢和你说话了。”

承认自己的不足，“我跟谁都没话说。”

拿事实提醒她，“刚还和你说话呢。”

企图蒙混过关，“《纸牌屋》你觉得怎么样？”

不耐烦，“你无聊不无聊呀！”

死猪不怕开水烫，“我就是这么个人，你还不了解吗？”

真诚地暴露自己的迷惑，“其实，我是不知道怎么接你的话。”

直截了当地告诉她，“我正忙着呢，别闹！”

.....

哪一种会管用呢？

哪种都不管用。

程序员是最能享受创造乐趣的人，也是最有创造力的人，可是面对“你为什么跟我没话说”这个两性话题，通常会手足无措，不知如何是好。

其实，我也是个不会和女人说话的程序员，我在这里写的、说的，多数不是个人经验或心得，而是臆想出来或者逻辑分析出来的，仅供参考。说到逻辑分析，多数程序员都擅长，当然，脑子里一边是水一边是面粉的除外。

**你的女人不断问你这个问你那个时，她们到底想干什么？**

**无它，求关注尔！**

这个时候你一定要放下自己的代码、游戏、网友，离开微信、微博、CSDN、cnblog、Twitter、Facebook，告别你的“魔兽世界”、LOL 或者“DOTA2”，迅速“穿越火线”来到伊人面前，清空脑袋里乱七八糟的代码逻辑、游戏场景，赶紧看看你的女人在干什么，想干什么，而你能在哪方面关注一下她。

比如女朋友在挑电视剧，其实她可能不是真的要看哪个特定的剧，而只是想你陪她一块看个什么，与你共度一天中难得的这两个小时的悠闲时光。要知道，**两个人在一起，最重要的其实是陪伴**，不管做什么，有你陪着，这就是一种幸福。所以，我们是挑《平凡的世界》也好还是看《天蝎计划》也罢，都不是关键问题。

比如你女朋友在化妆镜前捋饬那张你已经熟到记得(40, 35)位置有一颗麻星(-22, -43)处有一个暗斑的脸，突然略带感慨地说“有鱼尾纹了”，很可能她并不是真的在说“鱼尾纹”的问题。要知道女人是最擅长比喻、暗示、象征、反讽这些修辞手法的，有时她们对年龄也有着比男人更大的恐惧。

那么这时候你怎么说呢？



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

不管怎么说，首先你应该跑过去，仔细端详一阵儿（此时可以琢磨琢磨要说的话），然后再和她聊点儿什么。切记，不能下意识地立马搭腔。

比如你端详一阵儿后认真地告诉她你没看见。

比如你告诉她鱼尾纹不算什么，你会一直陪着她，和她一起上演最浪漫的事。然后给她讲讲杜拉斯小说《情人》，聊聊开篇的那段话——“那时候，你还很年轻，人人都说你美。现在，我是特意来告诉你，对我来说，我觉得现在你比年轻的时候更美。与你那时的面貌相比，我更爱你现在备受摧残的面容”。忽然发现这段话里面有个词儿不好，你还是慎重使用吧。

还有一些时候，你那位和你聊她单位的事儿，聊某个女伴的八卦，聊某个新闻，聊某个女星的人造胸器，其实都不是想聊这些具体的东西，**她们只是想和你说话，随便聊点什么**。这时作为程序员的我们，要积极回应她们，想办法让话题继续下去，而不是扮演“冷场王”的角色。

比如女朋友说《天蝎》很好看，你就可以和她聊里面的歌曲刷榜算法，聊黑客，聊网络盗窃，聊比特币的安全……反正乱七八糟什么都能聊，别说你喜欢上了某个餐厅女服务员就行了。

比如说女朋友喜欢看电视剧《平凡的世界》，你就可以吐槽佟丽娅演田润叶演得不像，李小明演田晓霞也演得不像，王雷的陕西话很蹩脚……很多，只要你能想到，还有你也可以和她讲路遥怎么写《平凡的世界》，聊和路遥同时代的陕西作家，比如贾平凹、陈忠实、高建群等的作品……这都很简单，作为程序员，你可能会了解各种推荐算法，比如基于统计的社会化推荐、相关推荐、协同过滤等，很容易就能应用到一部电视剧、一个明星、一个作家身上，反正沾边儿的都可以胡扯，即便你一时推演不出来相关内容，也可以参考各个视频网站、视频资讯网站等。把你攻克技术难题的劲儿分 5%出来，什么谈资都可以找到了。

## 自动过滤

不知道你有没有过这样的经历：

下班回到家里了，那个诡异的 Bug 依然对你恋恋不舍，系统的逻辑流程图忽隐忽现在你脑海里盘旋，虽然你忙着脱鞋、忙着响应女朋友的问长问短、忙着问女朋友这个那个，或者

忙着问孩子今天学了点什么、和谁玩儿了，可所有这些忙着，都是表面的、不自觉的、未经深思的、漂浮在你注意力之外的，当他们回应你或者你回应他们时，所有这些回应，也会像风中的那片云一般倏忽来去，你记得有那么回事儿，但具体是怎么回事儿却不甚了了。

举一个鲜活的例子。

妻子托人从德国代购了两盒小朋友吃的东西，因为她第二天有别的事情不能去朋友那里取，我就担当起了去朋友家取东西的重任。一盒三十八元，两盒七十六元。我骑着单车，穿透雾霾，往朋友所在的小区进发。快到时我居然不知道要给人家多少钱了，打电话给妻子，妻子说七十六，然后我又发现没有零钱，只好给朋友打电话，告诉她我没带零钱。后来的事儿没什么关系，不说也罢。

回到家里，媳妇很负责任地帮助我还原了我们之前的对话场景。她当时告诉我两盒七十六，还给我准备了零钱，问我要不要，而我告诉她不用，我钱包里有。到了今天，我不但钱包里没七十六块零钱，也不记得多少钱。

妻子发问了：“你最近老这样，脑子里在想什么呢？和你说的话，转眼就不记得了。”

我的话你听进去了吗？



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

其实都是因为我脑子里一直在思考工作相关的事儿，什么招人、代码、Bug、写文章，这些事情件件都如野马，我脑袋里就那么大点儿地儿，被它们的铁蹄子来回践踏，狼烟四起，别的事儿都被淹没了根本留不下痕迹。

诸如此类的事情，经常发生在我这个老程序员身上。妻子戏称我针对她设置了过滤器，自动过滤了她的各种消息。其实我真不是要过滤什么，实在是心有千千。

认真负责的程序员经常会陷入到某个情景中不能自拔，以致于对外界的其他刺激置若罔闻，别人就觉得我们像痴傻的“游魂”一样。

怎么破呢？

不是东风压倒西风，就是西风压倒东风。

就这么简单：中断、保存场景、还原。

具体怎么说呢，就是你下班前梳理一天的工作任务，把没有完成的任务以及对这些任务的思考线索记录下来（比如可以用 Evernote、有道云笔记之类的工具）。

最后一抹夕阳早已消失了，街灯已经全部点亮，钢筋水泥世界里的夜幕如此稀薄，你告别了依旧灯火辉煌的写字楼，走过车流明显稀少的街道，终于回到家里，开始面对你满腹牢骚的女朋友。相信我，之前你将工作场景压栈的动作绝对有用，因为你不用担心“某条重要的线索会不会被忘掉”这类问题了，所有参数已经入栈，明天到单位还原即可，所以呢，你就有了精力来拥抱扑面而来的各种问题。

## GoF 的设计模式

假如你是一个程序员，你一定知道 GoF 和他们的《设计模式》一书。什么，不知道？好吧，以后你会知道的。

设计模式（Design Pattern）是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结。我在我的博文“设计模式介绍之一：开篇概述”一文中说这是通俗的解释，现在看来还是有点晦涩了。

假如 GoF 总结的 23 种设计模式你都能理解并合理应用，那你应该也有能力理解与女朋友之间的“沟通模式”。

多数人说话、做事都有模式可循，比如我们老责备人的话——“哎你这人怎么这样啊”，其实静下心来想想，不“这样”他还就不是“你这人”了啊。人人都有习惯和行为模式，就看你能不能发现了。

有的人看你两眼就能看透你的很多行为模式，这些人我们称之为“先知先觉的半仙”。

有的人比较呆，比如我，和女朋友吵上多少次架都不长进，下次还会掉同一坑里，往同一个结果走。这些人“后知后觉”。我是程序员，以一斑窥全豹，我就觉得很多程序员也这样，不知道有没有道理。

还有一些人，根本是“不觉”，一个代码片段用一百次也不知道封装成一个函数，每次都拷贝、粘贴，空格和注释都不带变样儿的。

不管你是“先知先觉”还是“后知后觉”，都有救，放心吧，只要你拿出学习设计模式的精神头儿来研究和女朋友之间的对话、吵架等不分大小的琐碎事儿，你就能发现这其中暗藏的奥妙——你们之间的“男女沟通模式”，一旦你理解了那个，通过训练，就可以形成应激反应或条件反射。比如刚有吵架的苗头，你就能发觉，然后就可以从你事先准备的甜言蜜语库里检索数百条适用的，铺天盖地撒出去，扑灭那些将燃未燃的小火苗，将战争消弭于无形。

最后，我要特别声明：这里所提的一切策略，我都没有实践过，都是从我个人经验反推出来的。什么意思呢，就是说，我这么做效果很坏，我就认为那么做效果会很好。如果你想参考，请自担风险。

## 程序员和女朋友（女版）

有人说这个时代有两类人的老婆（女朋友）比较特别，应该要受人尊敬，第一类是军人，第二类是程序员。我是程序员，听到这话不知道该欣慰还是该忧伤。不管怎样，程序员确实有点儿特别，当他们的老婆（女朋友）还真的有很多需要注意的地方。下面就来说这个，如果你已经和程序员在交往，或者你已然与程序员结了婚，我要说的这些，可能会让你们琴瑟和谐。

### 明确提出你的要求

两个人在一起，开始总是惊心动魄，患得患失，随着深入了解，神秘感褪去，慢慢会失去新鲜感，激情归于平淡。也许不再为对方牵肠挂肚，也许不再芝麻大小的事儿都立刻要说给他听，也许不再一日不见如隔三秋，可是默契与信任会慢慢产生，与对方在一起，感到放松而踏实，激情虽然不如开始时澎湃，但深情却在一日又一日的平淡中积累起来。

两个人在一起，最重要的是陪伴。比如晚上或者周末的几个小时时间，你忙你的，我忙我的，看上去两个人都在自己的世界里，但你知道在这一片属于你们的空间里，你的那个他就在那里，你会感到踏实，这就是爱情，最常态的爱情。

相当一部分程序员都是很忙的（当然，我也见过一些很闲的，一星期不写一行代码，不画一张设计图），他们有时会被动加班，有时会主动加班，不加班了，他们还会自己琢磨点什么，学习点什么，即便回到家里，有很多人也会把自己绑在电脑前。总之他们的时间也是一天 24 小时，可是能分给你的就是不多。

你需要被关注，需要确认他是否在乎你，需要他陪你哭、陪你笑、陪你聊天、陪你逛街、陪你八卦，可是他就那么厮守着电脑，还一副心安理得的样子。你独坐在沙发上，怒火慢慢从隐约可见的小星星变得烈焰冲天，你几乎控制不住要脱口而出：“你能不能关了你那破电脑！”

哦，亲爱的，千万别这样。

他只是一根筋的“程序猿”，因为你说你有事儿要和他聊，他就觉得真没什么事，所以才开始玩电脑，温习一下白天用过的框架，深入学习一下，或者开始为自己的未来奋斗，努力想自己探索点什么积累点什么，或者——他真的在玩儿游戏。

好吧，无论如何，请记住，他是很在乎你的，只是他们是“人格残缺的高智商男性”，不单不知道怎么表达，也讨厌学习怎么表达。但是，他们真的是在乎你的。只是，没那么主动而已，不爱表现他们对你的在乎而已，他们会觉得：“我当然在乎你啊，你当然明白我在乎你啊”。

他们会觉得他们已经在陪你了一——可你的反应是：哦，天哪，这也叫陪伴！

你觉得不是，他们觉得是。很多程序员会觉得，我人在你身边，就已经算是把时间给了你，这就是陪伴了。如果你指责他们，他们会说：“我这不一直陪着你呢嘛。”如果你生气，对他们怒目相向，他们可能会不理解，可能感到你在无理取闹，可能会反唇相讥，也可能会含冤抱屈地回你一句：“你还想怎么样啊！”

要知道，程序员生活在确定性的世界里，总希望  $1+1=2$ ，总以为 `printf("Hello World!")` 就不可能输出 `"Goodbye World!"`。这种工作习惯会给程序员打上深深的烙印，他们特别容易形成“这事儿就是我想的这样”的思维习惯。

要知道，一般的程序员也习惯了“接受任务、完成任务、接受任务、完成任务……”这个工作模式，这也会深深地影响到他们的思维和习惯，他们会有这个反应：“你没告诉我我怎么知道呢？”

两下结合起来，在与你在一起时，我们这些程序员也经常是这么做的：想让我陪你你说嘛，你不说我怎么会知道？

在这种时候，不要生气，要淡定，要表现出你的善解人意，要主动聊他感兴趣的话题，要主动要求他们为你做事儿。此时你就是需求方，需要明确提出要求，他们要是不听，你就三分钟变化一下需求，直到他们心烦意乱再也没法淡定地坐在电脑前。

也许你觉得这简直不可思议，你难以接受居然要你一个大小姐主动求着那个不解风情的“阿猿”陪你说话聊天这个事实。

睡美人等待王子来解救的时代已经过去了，在倡导男女平权的时代，没有男人就应该主动、女人就应该被动这种定律，真心相爱的两个人，无论哪一方，都应该发自内心积极主动

## ☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

地珍惜彼此、珍惜在一起的时光。

感情需要经营，这绝对是两个人的事。

遇见不解风情的程序员，遇见不懂体贴的程序员，遇见情商低下的程序员，遇见自以为是的程序员……正是你一展风采的好机会，你一个投怀送抱他们就死心塌地了，所以呢，面对这些程序员，你需要明确提出你的要求，尽量让你的要求具体一些，这样他们的响应会更及时。比如你可以说，“跟我一起看会儿《何以笙箫默》吧”；比如你可以说，“咱们一起来看《天蝎》吧，里面有一群智商特别高的天才”；比如你可以说，“周六早上九点咱们去 XXX 羽毛球馆打羽毛球”；比如你可以说，“我没有春天穿的裙子了，明天下午咱们去 XXX 看看吧”……像这种言之有物的需求，程序员很少会忽视。当然你也可以态度明确地表明你的诉求，比如你可以说，“今天心情很不好，你过来陪陪我”；比如你可以说，“我很累，晚上不想做饭了，你带我出去吃西餐”……总而言之，你对他们的要求一定要具体，不能模棱两可空洞无物，这样他们久经训练的高智商大脑就会马上有反应，做或者不做，都会给你一个答复。

## 他送给你的礼物

每个女人都渴望浪漫和惊喜，平平淡淡的生活里，偶尔天外飞来一发幸福的子弹，在湖里荡起层层涟漪，这感觉，真不错。

比如你希望男朋友送你喜欢的那种味道的香水，或者送给你 99 朵玫瑰，或者能偷偷买件你看了几次却舍不得买的白色公主裙……不过，程序员很少会闻香识女人，你的希望很可能会落空。

下面是我的非典型送礼流水单：

- 妻子上学时我送她一个现代的 MP3，白色的，心形，我觉得好看。
- 西门子的手机，白色，我觉得好看。
- 三星的翻盖手机，银色，我觉得好看。
- 诺基亚的音乐手机，红色，我觉得好看。
- 有一次抽奖得了个 iPod shuffle，我挑了个红色的，送给妻子，我觉得好看。
- 小米 2S，刚出来时我在小米官网上抢的，送给妻子，我觉得好用。

- 中兴 Z5 mini，就是国母用过的那款手机，我给妻子买了一个，我觉得还成。
- 2014 CSDN 博文大赛我的奖品是联想 MIIX2 8 平板电脑，送给妻子了，这个糟透了。
- 2014 博客之星的奖品是 iPad mini 3，送给妻子了，这个不错。
- 华为的 P7，话说华为的手机，进步不小啊，越来越好了。

.....



你看到了吧，作为一个程序员，我送给妻子的礼物，以电子产品为主，很少有其他的東西。这是因为我喜欢电子产品，我认为把自己喜欢的东西送给女朋友是一件顶好的事情。但听说小猫请小狗到家里吃饭，准备了一大堆鱼，可是小狗却不高兴。

作为女生，你很可能希望你的男友送给你的小礼物，能让你有一点点动心，会让你情不自禁，可是你一定要有心理准备：程序员热爱电子产品胜过一切，他可能不会送你女士时尚



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

手表，而会送你一个 360 安全手环，这个手环可以用智能手机控制和监控，一旦你离开他超过 200 米就报警。

无论你的程序员男友送什么电子产品给你，他都是本着“把我认为最好的送给她，她一定会高兴”这种想法，你都应该及时表现出兴奋、惊奇、爱不释手的样子，这样会让他们有那么一点点得意，下次还会送你更好的电子产品。如果你脱口而出“怎么又是 XXX 啊”那就坏了，他的自尊心会受到严重打击，也可能会反应过度，脸一下子垮下来了，扔给你一句“爱要不要”然后就打开电脑玩游戏了。

也有一些程序员是挺浪漫的，可能会把你们在一起的照片集成在一起，做一款 App 送给你，或者为你做一个个人站点放上去，或者把你们一路走过的各种小欢乐、小悲伤画成漫画制成电子书送给你……当然，这都是程序员自以为的浪漫，也许你不屑一顾，但，请别拒绝这份心。

我想说的是，程序员虽然很少会制造浪漫，但他们却恰恰是最可能静静守候，不离不弃的伴侣；而那些浓烈如酒、疯狂似醉、浪漫满怀的人，却往往是醒来无处寻、来去都如风、最能往你心上留伤痕的人。

## 他喜欢什么呢

程序员喜欢什么呢？

作为程序员的女朋友，如果你爱他，你一定很关心这个问题。

其实，这个事儿太简单了：第一，他们喜欢电子产品；第二，他们喜欢电子产品；第三，他们喜欢电子产品。

你看我们程序员送礼物给女生都是电子产品居多，自然本人也是喜欢电子产品的。什么手机、手环、平板、眼镜、VR 头盔、盒子、游戏主机、笔记本、电纸书、MP3/MP4、无人机……只要是和电子产品沾边的，都喜欢。要是你的男友碰巧是搞开发的，却又对什么电子产品都没热情，那照我看来，他多半就是混到程序员队伍里的南郭先生。

除了电子产品，程序员还喜欢 IT 界的那些人和事，比如他肯定知道乔布斯和他的车库，

知道雷军和小米，知道佩奇和被墙的谷歌，知道库克出柜以及苹果，知道红衣大炮和 360，知道扎克伯格和脸书，知道特斯拉和马斯克，知道马化腾，知道刘强东和奶茶妹妹，知道马云和阿里，知道张朝阳和搜狐，知道黄章和魅族，知道王思聪和他爹，知道罗永浩和锤子，知道张小龙和微信，知道李开复和创新工场，知道马佳佳和她的情趣用品店，知道余佳文和超级课程表……凡此种种，只要你关注一个 IT 资讯网站，就可以了解到，然后就可以和他们瞎聊了。

至于 IT 资讯类的站点，你看看虎嗅（<http://www.huxiu.com>）、品玩（<http://www.pingwest.com>）、极客公园（<http://www.geekpark.net/>）、瘾科技（<http://cn.engadget.com/>）等就可以了。或者你想懒省事儿，那在百度上搜诸如“互联网、电子产品、IT 新闻”之类的关键字也可以。

程序员还喜欢游戏，LOL（英雄联盟）、WOW（魔兽世界）、愤怒的小鸟、传奇、梦幻西游、三国杀、德州扑克……从桌面游戏、网络游戏、手游、桌游一直到卡牌类游戏，都有人喜欢。

还有的程序员也喜欢研究股票、基金、期货、现货白银、现货黄金、余额宝、P2P 网贷……

总之，程序员的喜好有这么些特征：科技范儿、烧脑、炫技。

除了知道程序员喜欢什么，还要了解一些专业术语，比如 Bug、漏洞、敏捷、迭代、需求、送测、上线、版本发布、数据库、服务、下载、API、互联网、嵌入式、App……

一旦你关注这些内容，就很容易和你的程序员男友聊到一起。所以呢，你爱他，就要了解他的工作和爱好，这样就能有共同语言，就可以聊得来，他就会把你当成能一块扯的同道中人，久而久之，你们就水乳交融、琴瑟和谐了。

## 程序员带给你的福利

和程序员谈恋爱，还会有各种福利。

比如，工作环境所限，程序员身边没什么女人，他又很忙，所以你不用担心他花心，当然你也不用担心你花心被他发现——你和人幽会时他没准加班赶进度呢。

比如，程序员的另一重身份是神奇修理工，家里的电脑、电视机、空调、冰箱、插座、

## ☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

抽油烟机、手机、收音机、音响、拖地机器人、路由器、机顶盒、电磁炉、微波炉、面包机、蛋糕机等等乱七八糟的设备，他们都有办法折腾，没他们解决不了的问题。

比如，程序员吃饭一般都不挑剔，马路边的盒饭、煎饼果子、方便面、过期压缩饼干等等他们都能接受，在他们专注地和电脑谈情说爱时，你在他电脑边放一盒鱼饵他都能当花生米吃掉，所以呢，你完全不必担心自己的厨艺，他们不挑剔，就算遇到挑剔的，你就告诉他，“你嫌难吃老娘还不伺候了”，他们立马就吃得很香了。

比如，程序员一般不在乎穿着，一条牛仔裤可以穿五年，一件公司年会发的 T-shirt 就够他度过十个夏天了，所以呢，你完全不必担心他为了买服装而和你抢家里买衣服的预算，甚至你还可以借口给他买衣服拉他去商场陪你逛，因为他的衣服两分钟就可以挑好。

比如，程序员的耐受力很强，除了你删除他的代码这种事情，其他都能接受，无可无不可，所以，生活中很多事情，你都可以尽情做主，不必在意他怎么说，完事儿了告诉他一声就行，他就是不满意，多数时候也就来两句“嗯”、“这样啊”就过去了。

比如，程序员是孺子牛，吃的是草，挤出来的是牛奶，只要有口吃的果腹，有件穿的蔽体，他就可以很专注地投入到他喜欢的开发工作中，有的人八小时不用上厕所，有的人八小时也不喝一口水，有的人烟烧到手指了都不觉得疼，总之程序员非常好照顾，一点儿都不需要你女朋友费心，只要编程这种强需求能被满足，其他的东西对程序员来讲都是浮云。所以，和程序员谈恋爱极其省心，根本不用关心他吃喝拉撒穿，你只要三不五时地表示很喜欢他专注工作的样子就够了，如果你还能三不五时地告诉他“老公我觉得你能力真的好强啊什么技术到你这里都变得好简单”那就攻无不克战无不胜了，保准你的程序员男朋友心里乐开了花，把你奉为本世纪最理解程序员的贤妻。当然，他加班时你也要表示支持，说上一句“男人就该像你这样有事业心”。

## 生个孩子吧

在我 29 岁的时候，我的女儿像天使一般降临。我看着她从妈妈身体里出来，我听到了她第一声细嫩的啼哭，在日后的生活里，我陪着她吃喝拉撒玩，陪着她闹哭笑叫爬，我不想错过她小小生命里每一个独一无二的欢乐时刻。你邂逅了一个小生命，并且能够与她日日厮守携手前行，这是一件多么美妙、奇幻的事情！

当孩子以一声啼哭向这个世界宣告她的到来，程序员啊，你的生活从此天翻地覆多姿多彩，孩子为你打开了教授家的衣橱，带你进入纳尼亚的传奇世界，它带给你的，永远是你想象不到的。

自从我有了孩子，我感到这个世界与以往完全不同了，孩子重新定义了我的世界，包括我的工作。



## 更有爱

我从 2009 年开始做互联网机顶盒，我经常和我的队友们说，机顶盒就像我的孩子，我孕育了它，看着它出生，看着它成长，看着它走出闺阁飞入别人的怀抱，我担心别人怎么对待它、评价它。我对它的牵挂，一刻也未曾停止过，就像我对女儿的牵挂那样。

我要说的是，如果你真的有了一个孩子，你就会发现养育孩子与开发产品之间神奇的相似之处。

毫无疑问，你会爱你的孩子——不管他（她）是怎么样的小生命，你都会爱他（她）。这种爱是无条件的，发自内心的。否则，你绝对承受不了他（她）的哭，他（她）的叫，他（她）的闹，他（她）的歇斯底里，他（她）的莫名其妙（虽然这不常发生）。当你有了孩子，你才真正知道无条件的爱与付出是怎样的一种奇妙感觉，你想为他（她）做一切事：你想代替他（她）感冒发烧，你想代替他（她）出疹子，你想代替他（她）摇摇晃晃地走路，你想给他（她）一个只有欢乐没有忧愁的世界……

这种由衷的爱，不但会充盈在你和孩子之间，还会影响你的生活与工作，你的心情与心态会发生深刻的变化，平和、仁慈、喜悦、良善、忍耐，你比以往任何时候都更容易拥有这些，你会因此变得更容易爱别人、爱工作。

作为程序员，我会爱我所做的产品，不管它现在是好是坏，不管它将来走向何方，我会觉得，它是我的孩子，我有义不容辞的责任，让它变好，让它成为对别人有用的东西，让它能够解决别人的问题，为别人带来欣喜和安慰。这种爱，会让你更愿意花心思琢磨你在做的事情，也让你更乐意看到它的点滴变化，它每一次好的转变都会让你高兴，每一次迭代都让你充满期待，别人的每一个肯定都会让你收获作为家长的那种难以掩饰的欢愉。

一旦你开始爱你的产品，一切都会不同：你会时刻为你的产品着想，你希望你做的每一个决定，都能引导它走得更好；你会比以往更为愉快地投入工作；你会更加友好地与别人沟通，哪怕面对总是惹恼我们的产品经理与需求分析师，你也会因为想到“他们也想这个孩子更可人”而更容易平和地接受他们变来变去的需求；你会更爱测试 MM，因为她们指出了你的孩子的不足，让这个孩子变得更加完美。

## 时间管理

孩子会极大地牵扯一个程序员的精力，尤其在孩子未上幼儿园之前，你可能整天要围着孩子转，换尿布、拉屎、半夜喝奶、半夜端尿、发烧、感冒、拉肚子、抓握、翻身、爬、学步、做适合幼儿吃的饮食……假如夫妻双方都有工作，双方的父母又因没有时间或身体不好而不能帮忙照顾孩子，那真够你受的。要知道，没有人天生会做父母，即便你做好了迎接孩子的一切准备（买各种幼儿用品，看各种教育资料，咨询心理医生，请教亲朋好友……），当孩子真的来到世上后，你还是会发现自己懂的太少，有太多的突发状况让你手忙脚乱手足无措。是的，一定是这样的，每个孩子都是独一无二的，父母需要和孩子一起经历、学习、成长，你只能摸爬滚打跌跌撞撞地伴着孩子前行。

假如你是一个男程序员，你除了要照顾孩子，也要照顾妻子，尤其是坐月子，一不小心就会留下一些月子病。但是，你还有工作，而且作为程序员，你的工作需要你专注、高效，是繁重的脑力劳动。

在这样一种状态下，你会发现时间简直是太不够用了，你急切需要孙悟空脑袋后面那一撮毛，可惜你没有，你只能另外想辙：怎样才能更合理地利用时间。

我有一个同事，在孩子还小的时候，每天早上六点多就到公司了——因为他晚上不能加班，只能通过早起来赶工作。还有一个原因是，早上这段时间是整块的、安静的、少有人打扰的：孩子还在酣睡中，不会突然召唤你；同事们要么没起床要么在上班途中，不会问你这个问你那个；客户多半是八点半或九点上班，也不会这时打电话来抱怨各种问题……

现在我也深深理解了早上几个小时时间的宝贵，要知道，这都是生活赐予我们的——多么痛的领悟，哦不，应该是多么幸福的领悟。我经常五点多起床，好好享用早上这段珍贵的安静时光：可以写点东西，可以快速地写几行代码，可以调试一个 **Bug**，可以画一张时序图，可以梳理一下今天的待办事项列表……

除了早上的大块时间可以利用，你还会想其他办法让白天的时间能更有效。比如你会列清单，排优先级，挑最要紧的事儿来做；比如你会给每天定合理的目标，只完成特定的几件重要的事；比如你会想办法推掉无可无不可的会议……很多以前你不愿意想不愿意试的招儿，

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

此刻你都会迫不及待地要试试。要知道，十分钟对你来讲都是非常宝贵的，你的工作不能因为你有了孩子停下来。

就在这段时间内，你可能变得更加积极和高效，并且会形成很多非常棒的习惯，这些习惯会影响你接下来的工作和生活，反哺你的付出，让你的未来，呈现更多的可能性。

如果你被孩子弄得心烦意乱，在工作上敷衍了事，那工作上的事儿则会反过来报复你，让你更加意乱心烦焦躁不安。微笑会带来微笑，苦恼会滋生苦恼，所以，不管怎样，记得我们会因此成长，这是我们的光辉岁月！要知道，光辉岁月不是指成功后那些春风得意马蹄疾的日子，而是那些我们勇敢面对奋力挣扎却永不放弃、在极度的苦闷中因幸福而含泪微笑的不朽历程。

好吧，如果你遇到了困难，也可以看看《自控力》、《高效能人士的七个习惯》、《不抱怨的世界》等书籍，它们会帮助你顺利走出泥沼。

## 退一步海阔天空

养育孩子就像万里长征，任重道远，要花的时间和精力难以计量，并且，这是一场没有终点的旅行，此爱绵绵无绝期。我们用于编程的时间明显少了，当你终于有时间坐下来准备写上一段代码时，又会有各种各样的事情像打地鼠一样蹦出来打断你。

你的时间不再只属于你，不再能够从心所欲地支配，有一个可爱又可气的小天使，随时都可能施展出召唤神兽的绝技把你从工作现场拐走。作为程序员，你将面临前所未有的挑战，如何管理时间成了非常迫切的问题。不过，前面我们已经提到了妥善利用时间的一些方法，正如鲁迅先生所说，“时间就像海绵里的水，只要你愿意挤，总是有的”。

孩子除了让程序员的时间碎片化，还会带给我们另外一种好处，真的，你没看错，是好处。

当我还没有女儿的时候，甚至在我妻子挺着大肚子的时候，我还一心扑在开发上，如有需要，会毫无顾忌地把绝大部分精力都投入到项目上。当遇到问题或者难题的时候，我第一反应就是迎难而上，不达目的誓不罢休，周末、工作日的晚上，统统都可以加班，我会逼着自己，哪怕历尽千辛万苦呕心沥血也要克服难题。

可是现在，我有了亲亲的小宝贝，她可能随时随地需要我，我不能那么任性了。有时你半夜三更苦苦思索毫无进展、焦虑不安时，女儿会用她一声稚嫩的啼哭把你从沉思中唤醒，啊，她要尿了！好吧，也可能是其他的事儿，比如她被饿醒了，你得给她冲奶粉。当然也可能是她做了梦，抓你的手没抓到惊醒了。总之，她有各种各样的办法占有你。

虽然我的开发被迫中断，可是问题依然在脑子里盘旋，在我摇晃奶瓶的时候，在我给女儿换尿不湿的时候，在我握着女儿的小手轻而又轻地在她身边躺下时，我的脑袋，神奇的脑袋，还在思考遇到的难题，我的大脑和思维，依然沉浸在问题中，它们开始用无意识的方式思考。

你离开了问题的现场，去做别的事情，但你还在想着问题。当你正在做着别的事情时，问题突然有眉目了，答案像剪径强盗那样突然蹦出来了。

多么神奇！

你有睡梦中突然惊醒，干掉 Bug 的经历吗？

你有吃饭时突然找到线索的经历吗？

你有拎着“妈咪宝贝”等待交款时灵光闪现的经历吗？

你有和女儿压着跷跷板忽然茅塞顿开的经历吗？

.....

我远离了工作，可是有时居然能找到新的办法，三下五除二就解决掉纠缠你几个小时的问题。

这给了我新的启示，也让我改变了面对难题的策略：不再一直和问题纠缠，尝试退一步。退一步会让你从只见树木不见森林的窘境中跳出，也许你转眼一看，忽然就发现海阔天空，那边有条新的道路。我经常会因此而有“山重水复疑无路，柳暗花明又一村”的感觉。

孩子逼着我学会并主动应用新的战略战术来战胜开发过程中的各种拦路虎，这真是意外之喜。



## 沟通

孩子的世界与成人不同，对事物的感知和看法也与成人不一样，他们有自己的方式，与大人完全不同的方式。并且这种方式，是我们大人早已经忘却了的、很难再俯拾起来的。所以，我们经常和孩子产生矛盾。

这种孩子和成人之间认知、感受方式上的不同，也同样存在于不同的成人个体之间，我们在学习如何与孩子相处的过程中，也可以学会如何与其他成人更好地沟通。

## 承认与接纳

我们经常否认孩子的感受，企图用大人在话语、身体、权威上的优势粗暴地压制住孩子，最终，我们和孩子的对话通常会演变成争吵。

比如要出门，孩子说热，我们说天气凉，非要孩子多穿一件衣服。

比如孩子周末早上睡起来没两个小时，到出门去上英语课的时候说自己困了，我们说刚起来怎么会困。

比如孩子说很讨厌某某老师，我们说不许这样说话。

比如孩子说青菜不好吃，我们非要说青菜好吃千方百计要把青菜塞到孩子嘴里。

.....

有太多的时候，我们否认了孩子的感受，用大人想当然的方式和成人世界里经过训练的规则否认孩子的感受，而实际上，孩子的感受是真实的。

的的确确，他们的感受是真实的，不掺杂一点儿做作。而我们却毅然决然直截了当地否认了他们，这会给他们带来困惑和愤怒。

在实际的工作中，我们也常常用这种方式对待我们的小伙伴。

比如张三觉得李四要做的 APK 自动升级很简单，认为他说 32 个小时才能完成包含了非常大的水分。

比如有新人不能理解成员初始化列表，我们这些老手很可能粗暴地说出“这是成员初始化列表”这种正确的废话。

比如小五负责的断点续传功能一个星期了还没完成，赵六就会觉得他在偷奸耍滑，认为小五总是说里面有很多困难一定是在为自己的懈怠和拖沓辩护。

比如测试 MM 说你提交的版本没有修复 JIRA 上编号为 9527 的 Bug，你通常会来一句“不可能，在我这里好好的！”

比如用户说升级新版本后软件打不开了，很多程序员的反应是“这个白痴根本不知道怎么用”。

.....

我们习惯了从自己的经验出发，惯于否定别人的话、别人的努力、别人的感受、别人的技术水平、别人对困难的感觉，基本的逻辑就是“我觉得这样，你也应该这样”。

可这是错误的。

孩子说他热，一定是热，我们不是孩子，怎么能认为他不热呢？

孩子说他困，一定是困，我们不是孩子，怎么能认为他不困呢？

小五说断点续传很难，那是基于他现有的技术水平，也许他连网络编程的概念都还没有，也许他从来没有使用过 `socket`；而赵六呢，一直在做网络方面的开发工作，`HTTP` 协议都实现过多次了。

用户说软件有问题，那一定是他遇到问题了，没问题谁会跑来骚扰你呢？你又不是声音甜美的客服 MM。

.....

因此，当别人向我们说出他的遭遇和感受，我们首先要承认他的感受，接纳他的感受，然后才有可能在接纳的前提下基于个体的差异性来达成正确的认识和沟通。

比如孩子说热，而我们觉得冷，怎么办？我们可以说，“我觉得冷，但你觉得这里热，

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

那你可以试试脱掉外套感觉一下”。就这样，承认他的感受，让他自己找平衡点。

比如小五说 HTTP 断点续传很难，需要两个星期才能搞定，怎么办？赵六可以说，“你觉得很难，一定是有很多想不通的地方，你可以把你认为的难点列出来，咱们一起来讨论一下。”

如果你否认孩子的感受，他会和你对着干，他哭，他闹，他别过头去不理你，他赖在地上不起来，他满地打滚哇哇乱叫……他控制不了自己的愤怒，他用自己的方式与你抗争，你很多时候会妥协，因为你爱他，你不愿意看着他“糟践”自己。但如果总是这样，你就会觉得哪里出了问题，这是你和孩子之间的事情，你爱你的孩子，你希望找到一种更好的沟通方式来避免这种往复循环的较量。于是你开始学习，于是你找到了《如何说孩子才会听怎么听孩子才肯说》，如果你坚持实践，你会发现，你的问题解决了。

那工作中呢？其实，很多时候你是没有那么多耐心去找更好的方式，你会习惯性地运用已有的从自我经验出发但是简单粗暴的方式来理解你的队友。但是，你不是他，你站在自己的角度看问题，你真的理解不了。如果你是领导，可能会利用职位上的优势和经验上的优势暂时压制住你团队成员对自己感受的诉说，你告诉他“下周三一定要出结果”，中间过程你不管。当然你可以这么做，但这不是最好的办法，小五可能含冤带屈地转身去工作，甚至会一腔怒火，这些不良情绪会让他的工作效率大大下降，如果他不能摆脱这种感受，那他要完成这项任务，会花费更长的时间，可能是三周或者四周。

如果你已经能够承认、接纳、理解你的孩子的感受，这些经验会帮助你，会让你尝试着站在小五的角度来思考他遇到的问题，那样你可能会认为小五真的有困难，你就会采取有效的措施让进度符合预期。

## 让赞赏飞

想想我们平常怎么赞赏孩子吧。

如果你没有孩子，就想想小时候你爸妈怎么赞赏你吧。

孩子今天自己穿上了套头衫，自己刷了牙，你说，你真棒。

孩子晚上画了幅画，色彩鲜艳，你说，你真棒。

孩子数数能数到 20 了，你说，你真棒。

孩子剥开一块大白兔奶糖，把糖纸扔到垃圾篓里了，你说，你真棒。

.....

孩子终于麻木了，“你真棒”这三个字，并不是灵丹妙药。因为，它们太不具体了。

要赞赏孩子，一定要具体，要描述他做了什么，要表明你肯定的是他哪个行为，唯其如此，他才能保持这个行为，下次还这么做。

当孩子完成了一件值得赞扬的事情，我们要用描述性的语言具体地指出某一个最突出的点，这种赞赏方式更容易让孩子接受，更容易强化效果。那种不管三七二十一统统“你真棒”的方式，会让孩子感到迷惘和困惑：我到底哪儿做得好呢？

其实我们在工作中也是一样的，每个人都希望别人夸自己，工作干好了，都希望别人能看到，你的代码干净、漂亮，逻辑清晰，一定希望别人能知道这些具体而微的点，那种“代码写得不错啊”之类的话，很可能引不起你丝毫的兴趣和好感（第一次听到除外），你可能会想：这家伙，根本就没看过我的代码！

那么作为程序员，我们真的想夸赞我们的同行时，一定要具体而微，把我们的赞赏落实到一个具体的点上。比如你发现小五写的断点续传模块容错性很高，你就可以说：“哇，小五，拔路由器网线这种奇葩情况你都考虑在内了，异常处理堪称完美啊。”

又比如李四的 APK 自动升级完成了，你这么说：“哇，李四，我发现你在下载 APK 包时使用了断点续传，可以大大节省用户的流量，真是想用户所想。”李四也许会脸红，但心里一定很乐，对你的看法也会不同。

.....

因此，如果你用心与你孩子相处，你就会发现描述性的赞赏效果显著，那你就会尝试着把这种方式应用到你的工作中来，你接着又会发现，咦，我的人缘怎么好起来了呢？

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

## 让他去试

孩子学走路，会扶墙，会扶茶几，会扶桌子，累了会接着爬，今天会走了，摔了一跤，几天都不敢走，当爸妈的该急了，学步带，学步车，张开双臂匡护它就像母鸡护小鸡。

孩子穿鞋，半天系不上鞋带，妈妈急了，一伸手给系上了。

孩子想拿筷子夹土豆丝，妈妈怕洒桌子上，娴熟地夹了一筷子送孩子嘴里了。

.....

太多的时候，我们“替”孩子做了选择，“替”孩子办了某件事，而这些，原本是他们要自己去感受，自己去完成，自己去体验自身能力的。我们的包办，会让他们丧失锻炼自己的机会，会让他们延迟独立，甚至让他们永远难以独立。

幸运的是，有人教我们不要这样，有人告诉我们要让孩子自己经历各种问题带来的挣扎，让他们在自己的错误中得到成长。我们爱孩子，所以容易接受这样的建议。

可是工作上呢？

能力强的程序员经常会不耐烦，直接就告诉后进人员这个事儿这样做那个事儿那样做；项目经理会圈定正确的方法，不想让团队成员走弯路。这样做从项目进度的角度来讲，无可厚非，可是从人员成长的角度看，会延迟团队成员的成长。每个程序员都需要经历自我摸索、独自寻找答案的过程，唯其如此，才能获得宝贵的经验，只有在一次又一次的摸索中，才会找到自己的学习模式，培养并提高自己解决问题的能力。而我们把答案直接摆到某个程序员面前，就貌似友好地剥夺了他成长的机会。

我们不愿这样对待自己的孩子，也请不要这样对待我们的工作伙伴。

假如有人来请教你问题，不要跑过去直接坐到他凳子上替他写代码。

假如有人来问你股票数据和界面呈现该怎么设计，不要直接告诉他答案，让他去看 MVC。

假如有人痛苦地来找你帮他解决 Bug，即便你早有答案，也不要直接揭开谜底，请问他

一些问题，尝试引导他自己分析为什么会这样，帮助他自己找到答案。

.....

他如果着急，可能会在心里嘀咕你，可他如果是个上进的伙计，会感激你的。

## 观察与猜度

由于孩子在 4 岁之前很难用语言准确描述自己的想法、感受、需求，我们不得不仔细观察他们，通过他们的动作、表情、情绪等来猜测他们的实际诉求，这会让我们变得更有眼力，能够看见行为背后的目的，会让我们在听不懂孩子的话的情况下也能理解孩子。这就是为什么没有孩子的人很惊奇为什么妈妈们听到孩子说肚子疼就知道孩子可能是要拉粑粑的原因。

我女儿上幼儿园，早上有时会不愿意起床，即便被叫起来了，也磨磨蹭蹭挑衣服，这件刚穿过，那件不漂亮，第三件又不搭配，好不容易衣服好了，又会说鞋子的事儿，鞋子和衣服配不到一起啦，鞋子夹脚啦……经过不断的观察和总结，我知道她在这样的早晨，是想通过这样的方式告诉爸爸和妈妈，她昨天在幼儿园遇到了不开心的事情或者担心今天会再遇到不开心的事情——比如中午睡不着觉之类的，一旦我们看到这个原因，就可以和她沟通了，帮助她摆脱担忧。

我们被迫炼就了火眼金睛，这个过程是痛苦的，但也是很有成就感的。

愿我们花费同样的精力在我们的同事身上，愿我们像爱孩子一样爱我们的工作伙伴。

比如你发现“某猿”最近代码产出很少，天天面色灰暗，那估计是遇到什么事情了。

比如“某猿”和别人谈笑风生，跟你原本也是有说有笑，可是最近老是公事公办的样子，闲言没有一句。

你因为孩子而养成的观察和揣摩的习惯可以帮上你的大忙，你有可能洞烛神机，及时发现你的某个行为可能伤害了别人，让你来得及挽回、修复彼此的关系，让配合继续下去。你也可能适时发觉某个伙伴生活上遇到了难解决的问题，能够在必要的时候给予他安慰，让他感到不那么孤单。

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

孩子是一面镜子，一面拥有魔力的镜子，能够折射出我们成人世界许多习以为常的荒诞。

当我们有了孩子，对孩子由衷的、无私的爱，会让我们发现自己身上形形色色的不健全，会让我们心甘情愿改变自己，会让我们更爱学习，会让我们更平和，更温暖，更容易为别人所接受。

我们因为孩子而养成的好习惯同样适用于成年人的世界，你周围的世界，会随着你的改变而改变，我们拥有了更多的可能性，你会看到前所未有的奇观。

## 推荐给程序员的书

程序员的世界里代码、设计、工具、硬件、数据等技术性内容，并且这些内容十足十的重要，它们影响着我们的工作、生活、人生价值，但还有更重要的事，那就是认识自己、发现自己。

人生最本质的财富，是你自己，你自己就是一座巨大的矿藏，只要开发，就能有无穷的潜力。也只有开发，你的一切才能显现出来，才能熊熊燃烧，才能闪出光彩。而这一切的前提，就是认识你自己。

要想认识自己，除了要与孤独、沉思为伍，还少不了集结了前人智慧的书籍。我曾看到一些给我教益的书，在这里推荐给大家。

### 《这辈子，我最想做的事》

这本书的英文名字更能表达作者的意思：A LIFE AT WORK ( The Joy of Discovering What You Were Born to Do )。我觉得翻译成“这辈子，我最想做的事”丢失了原文的一部分韵味。

美国摇滚巨星斯汀（Sting）说：如果我早些看到这本书，我可能提前三至五年取得现在的成就。

纽约时报评论说：这是一本了不起的书，让人认清工作真正的意义。

这本书的作者是托马斯·摩尔（Thomas Moore），他曾经是修道士、音乐家、大学教授与心理治疗师。他是多中心生活的倡导者，他经历了各种身份，最终认为通过写作、演讲、心理咨询等活动引导别人认识自己、发现自己、活出自己是他的终身工作。他在这本书里，告诉我们如何感应来自心灵的召唤，如何找出隐藏在平凡中的神圣，如何找寻我们生命中最需要的东西。

托马斯·摩尔（Thomas Moore）的这本书告诉我们，“**成功是用自己的方式度过人生**”。他通过这本书引领我们穿越世俗的成见与泥沼，教会我们怎样找到承载与体现人生意义的“终身工作”。



☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

你的工作是自己喜欢做的事吗？你是否觉得工作特别累？你是否经常为工作而烦闷？你是否总是觉得你的工作没有意义？你是否想换工作却又不知道什么样的工作适合自己？……

你需要这本书！

## 《少有人走的路》

“少有人走的路”系列共有《少有人走的路 1:心智成熟的旅程》《少有人走的路 2:勇敢地面说谎》《少有人走的路 3:与心灵对话》《少有人走的路 4:心灵地图》四本。前三本的作者 M·斯科特·派克是“我们这个时代最杰出的心理医生”，治愈了成千上万的病人。“心灵地图”的作者也是上文中提到的托马斯·摩尔（Thomas Moore）。

“少有人走的路”系列被西方媒体誉为“来自上帝之手”的时代杰作，创造了出版史上的一大奇迹。

这套来自异域的书，居然揭示并击中了我内心的秘密，我那些迷惘、挣扎、困惑、痛苦都被它照亮了，它给了我很多安慰和启迪，我会一读再读，它将伴着我穿越错综复杂的人生沙漠。

我唯一感到遗憾的是：为什么我没有早一天遇见它！

生活就像一面镜子，你对它笑，它对你笑，你对它哭，它也对你哭。从某种意义上讲，你面对生活和工作的态度，决定了你能拥有什么样的人生。

我很庆幸我曾经读过下面的两本书，我的生活已经因为它们发生了不小的变化。

## 《致加西亚的信》

谁能把信送给加西亚？

为什么罗文可以把信送给加西亚？

你是不是罗文？

你要不要做罗文？

《致加西亚的信》已经畅销了 100 多年，各种版本累计销售量近 8 亿册。2000 年被美国《哈得森年鉴》和《出版商周刊》评为有史以来世界最畅销图书第 6 名。当然这绝不是我推荐它的理由。

我是在一个偶然的的机会幸会《致加西亚的信》的：女儿在上课，我有时会在专设的休息区等候，那里有一个书架，摆满了由学校校长亲自挑选的书，我在浏览时看到《致加西亚的信》这本薄薄的小册子，翻了几页后立刻被吸引。

每一次相遇都是缘分，有些缘分是一辈子，有些却只是短短一瞬，我认为与《致加西亚的信》的缘分，可以持续一辈子，它所倡导的敬业、忠诚、勤奋的思想观念深入我心，也是我一直以来所推崇的。最重要的是，它唤醒我们，使我们再一次意识到，你所有的敬业、忠诚、勤奋、高效、执行力……都是为了自己的将来。

世界上大多数人都在为薪水而工作，如果你能不为薪水而工作，你就超越了芸芸众生，也就迈出了成功的第一步。

请谨记你是在为自己而工作，谁都无法阻止你为将来所做的努力，谁都无法剥夺你因此而得到的回报。

## 《不抱怨的世界》

来自诗人马雅·安洁罗关于抱怨的智慧之语：“如果你不喜欢一件事，就改变那件事；如果无法改变，就改变自己的态度。不要抱怨。”

《不抱怨的世界》把上面的话放在前言的前面，它直截了当地道出了我们面对不满意的事件、工作、生活时应取的态度，而威尔·鲍温则从概念、理论和逻辑出发深入阐述了不抱怨的世界，同时还设计了一套易于执行的、行之有效的、挑战抱怨的修炼方法，你只需要 1 只紫色手环和 21 天，就可以改变自己的命运。

一直以来我以为自己是很少抱怨的人，可是当我看了《不抱怨的世界》之后，发现自己在日常的工作与生活中还是充满了抱怨，于是我决定带上紫手环，试图度过充满挑战的 21 天，连续的、不抱怨的 21 天。紫手环的用法是这样的：一旦发现自己抱怨，就将手环从左手

☞你好哇，程序员——漫话程序员面试求职、升职加薪、创业与生活

换到右手，再发现自己抱怨就将手环从右手换到左手。

我没能持下来，但我认为我的尝试依然对我产生了深远的影响：我发现自己的抱怨比以前又少了一些。

抱怨是最无意义的消耗能量的活动，看了此书，哪怕你只能在数天之内减少抱怨，也是值得的；而如果你能坚持 21 天不换手环，养成不抱怨的习惯，那你的人生将会因此不同，神奇美丽的新世界会施施然向你走来并拥抱你。

要想成为一个优秀的程序员，除了能够摆平计算机搞定 Bug 之外，还要有良好的沟通能力。沟通能力上的差别，是决定你能否走得更远的关键。

作为一个程序员，我也是内敛、木讷、羞涩、惜字如金的，而且从心里会对巧舌如簧爱耍嘴上功夫的人不屑，不过，生活处处有学问，有口才的一定是人才，这是我们不能视而不见听而不闻的。

当然，沟通也不仅仅是“说话”，它只是在更多时候通过“说话”这种方式表现出来。如果你要了解怎样才能提升沟通能力，我可以从非专业的角度推荐两本书。这只是我认为好的，但愿也适合你。

## 《如何说孩子才会听怎么听孩子才肯说》

- 一本融合爱与沟通技巧的神奇之书
- 一本已让全球数百万家庭收获亲密亲子关系的家教图书
- 一本持续创造销售奇迹的家教图书
- 一本让孩子健康成长的父母必读之书

这是《如何说孩子才会听怎么听孩子才肯说》的封面内折页上的介绍，诚如你所见，它首先是一本育儿类的家教图书。我也是因为有了孩子才知道这本书，但当我拿到书读了几章后就发现：我真应该早一些拥有它！

《如何说孩子才会听怎么听孩子才肯说》不仅提供沟通的技巧、说话的策略、解决冲突的思路、赢得合作的方法等等操作层面的指南，还告诉我们应该这么做背后的深层原因，而

这一切，同样适用于成人世界。

如果要我推荐教人沟通的书，它必然是第一本！

## 《蔡康永的说话之道》

这本书的短文篇篇精彩，每篇谈及一个特别的方面，轻松的语言和故事让人有种一览为快的感觉，你不会觉得自己是在刻意去学、去练某个技巧，你感觉他在和你八卦别人的家长里短而你作为观众却也受到了启迪。每篇短文前还有搞笑的熊宝漫画，与文章相映成趣，也是一大特色。

这本书我通览一遍，现在还不时翻一翻，它于轻松、诙谐之中谈论说话的技巧，由术入道，确实很妙。

或有疑问：既然是“推荐给程序员的书”，为什么不推荐一些专业技术方面的书籍？

这是因为，一方面技术方面的书籍很多，很容易找到，在网络上随便搜索一下，就可以看到各式各样的书单；另一方面，我始终认为，“功夫在诗外”，对于个人成长来讲，往往那些看似无关要旨的事儿却恰恰是能够从内而外改变人、影响人的关键，并且，很少有人告诉我们这个重要的事实。